Euskal Herriko Unibertsitatea / Universidad del País Vasco

Lengoaia eta Sistema Informatikoak Saila

Ikasketa automatikoan oinarritutako gaztelaniako perpaus-identifikatzailea

Andoni Ibirriaga Godoyk

Informatikan Ingeniari titulua eskuratzeko aurkezturiko

Proiektua

Zuzendaria: Bertol Arrieta Kortajarena

Donostia, 2012ko maiatza.

Laburpena

Proiektu hau gaztelaniarako perpaus-identifikatzaile automatiko bat garatzean datza, ikasketa automatikorako FR-Perceptron softwareaz baliatuz. Aplikazio hau aurrera eramateko, Ancora corpusa eta Freeling analizatzailea erabili dira. Ancora corpusetik, perpaus hasieren eta amaieren informazio zuzena (eskuz etiketatua) atera da. Era berean, Ancorako jatorrizko testua, Freeling programarekin analizatu da. Izan ere, ikasketa automatikoan lortutako emaitzak ahalik eta errealenak izan daitezen, ikasteko erabiltzen den informazioak automatikoki lortua izan behar du, eta ikasi nahi den horrek, berriz, zuzena eta beraz eskuz etiketatua. Corpusa hiru zatitan banatu da eta zati handienarekin, *FR-Perceptron* programa erabiliz, informazio morfosintaktiko gehiago edo gutxiago emanda, hainbat proba egin dira, jakiteko zein informazio linguistiko den esanguratsua ataza honetarako. Ancora corpuseko informazio linguistikoa erabiliz ere hainbat proba egin dira, informazio zuzenagoa izanez gero gure aplikazioak lortuko lituzkeen emaitzak aztertzeko.

Hitz gakoak:

1. Hizkuntzaren Prozesamendua
2. Azaleko sintaxia
3. Perpaus-identifikatzailea
4. Ikasketa automatikoa
5. Perceptron

Gaien aurkibidea

Irudien zerrenda

Taulen zerrenda

I. KAPITULUA

Sarrera

Proiektu honetan gaztelaniarako perpaus-identifikatzailea aurkezten da, testu bat emanda, bertan dauden perpausak identifikatzeko gai den sistema.

Azaleko analisi sintaktikoaren baitan kokatzen den eginkizun bat da perpausen identifikazioa. Oso baliagarria izan daiteke Hizkuntzaren Prozesamenduko zenbait aplikaziotarako. Adibidez, testu baten analisi sintaktiko sakonagoa egin nahi bada, analisi horren lehen pausoa izan daiteke testuko perpausak identifikatzea. Baita ere, oso erabilgarria izan daiteke itzulpen automatikoa erabili nahi den kasuetan, itzuli beharreko zatiak perpausetan banatzen badira, perpaus hauek itzuli beharreko unitate bezala erabil baitaitezke.

Perpausen identifikazio automatikoaren aurretik kateen identifikazio automatikoa landu izan ohi da Hizkuntzaren Prozesamenduan. Kateekin izandako emaitza onek eraman dituzte perpausen identifikazio automatikoan sakontzera arlo honetan lan egiten duten ikertzaileak. Azken finean, hitz kateei aplikatzen zaien hainbat teknika oso baliagarriak izan daitezke perpausak lantzeko garaian. Kontuan izan behar da, dena dela, perpausen identifikazio automatikoa kateen identifikazioa baino zailagoa dela, kateek ez baitaukate izaera errekurtsiborik, bai ordea perpausek. Beste modu batean esanda, kate bat ezin da beste kate baten barruan egon; perpaus bat, aldiz, beste perpaus baten barruan bai.

Arrieta (2010) lanean oinarritu gara proiektua aurrera eramateko. Izan, ere euskararako perpausen identifikazioa landu da bertan.

Gure proiektuaren helburua ikasketa automatikoa erabiliz gaztelaniara-

ko perpaus-identifikatzaile bat sortzea da eta lortutako emaitzak aztertzea. Ikasketa automatikorako pertzeptroiak erabiliko dira eta horretarako, jada sortua dagoen tresna batez baliatuko gara, aurrerago azalduko dugun *FR-Perceptron* softwarea. Proiektuaren zati garrantzitsu bat Ancora corpusaren eta Freeling analizatzailearen irteeraren formatua aldatzea izango da, ikasketa automatikoko programak uler ditzan. Horretarako, guk behar ditugun datuak atera beharko dira XML formatuan dauden fitxategietatik. Freeling analizatzaileak emandako informazioarekin hainbat proba egingo dira ataza honetarako informazio linguistiko esanguratsua zein den ebazteko. Gainera, Ancora corpuseko informazio linguistikoa erabiliz ere hainbat proba egin dira, informazio linguistiko zuzenagoa izanez gero gure aplikazioak lortuko lituzkeen emaitzak aztertzeko.

II. KAPITULUA

Aurrekariak

Aurkeztutako arazoari emandako erantzuna azaldu aurretik, lan honetarako erabili diren baliabideak aurkeztuko ditugu, eta baita lan honen oinarrian dauden ikerkuntza-lanak ere. Bai gaztelaniaz bai katalanez aurki daitekeen eta notazio maila desberdinak dituen corpusa da Ancora. Hona hemen zenbait notazio maila:

1. Lema eta kategoria morfologikoa.
2. Osagaiak eta funtzio sintaktikoak.
3. Egitura argumentalak eta gaikako funtzioak.
4. Ahozko klase semantikoak.

Gaztelaniako corpusak 500.000 token baino gehiago ditu eta prentsako testuetatik lortutako esaldiak dira gehienbat. Notazio prozesuaren ondorioz, aditzen lexiko bat ere sortu da 2.647 sarrera dituena gaztelaniako bertsioan (Ancora web orrian[1] dago eskuragarri).

Corpusa, XML formatuan dator, eta gure proiektua aurrera eramateko formatu aldaketa bat egin beharko da, aurrerago azalduko den bezala, eta bertatik ikasketa automatikorako interesatzen zaigun informazioa atera. Ancora corpusa nolakoa den hobeto ikusteko hona hemen zati bat, adibide gisa:

Informazio ugari atera daiteke corpus honetatik; hala nola, guri interesatzen zaizkigun atributuak: forma, lema, kategoria, azpikategoria, katea eta

[1] http://clic.ub.edu/corpus/es

```
<sp>
  <prep>
    <s lem="a_partir_de" pos="sps00" postype="preposition" wd="a_partir_de"/>
  </prep>
  <sn entityref="ne" ne="date">
    <grup.nom gen="m" num="s">
      <w lem="[??:??/??/1994:??.??]" ne="date" wd="1994"/>
    </grup.nom>
  </sn>
</sp>
```

Irudia II.1: Ancora jatorrizko corpusaren zati bat.

perpausen mugak. II.1 irudian, *gen* atributuak hitzaren generoa (masculino, femenino) adierazten du; *lem* atributuak, berriz, hitzaren lema zein den; *num* atributuak singularra edo plurala; *pos* (Part of Speech) atributuak kategoria adierazten du bertan agertzen den lehenengo hizkiaren arabera; *postype* atributuak azpikategoria adierazten du; *wd* atributuak, azkenik, hitza testuan zein formatan agertzen den adierazten du. Haipatutako irudian ikus daitezkeen *sn*k eta *sp*k, izen-sintagmaren eta preposizio-sintagmaren mugak adierazten dituzte hurrenez hurren.

II.1 Freeling

Freeling hainbat hizkuntzatan idatziak dauden testuak automatikoki analizatzeko kode irekiko liburutegi bat da. Freelingek Hizkuntzaren Prozesamendurako azaleko analisia eta testuen notazio linguistikoa eskaintzen die software-garatzaileei; honek garapen kostuak asko jeistea ahalbidetzen du. Honez gain, Freeling oso moldagarria da eta sendotasun nahiz azkartasun aldetik kasu errealen aplikazioetara zuzenduta dago. Garatzaileek defektuz dauden baliabide linguistikoak molda ditzakete beharrezkoak dituzten domeinuetara, eta baliabide berriak gara ditzakete hizkuntza bakoitzak dituen ezaugarri berezietarako.

Gure kasuan, proiektu honetan lortutako emaitzak ahalik eta errealenak izan zitezen, Freeling erabiltzea erabaki genuen testuaren azaleko analisi morfosintaktikoa aurrera eramateko.

II.2 CoNLL 2001 biltzarra

Proiektu hau ezingo litzateke bere osotasunean ulertu CoNLL 2001 biltzarreko ataza partekatua kontuan izan gabe.

CoNLL-2001 biltzarreko ataza partekatuaren helburua perpausak identifikatzea izan zen. Ataza partekatu honen bidez, metodo automatiko desberdinak ebaluatzen saiatu ziren, batez ere ikasketa automatikoko metodoak.

Hau egin ahal izateko, formatu berezi bat definitu zen, gure proiektuan *CoNLL formatua* deituko dioguna. Formatu hau hainbat zutabetan eta hainbat lerrotan antolatzen da. Lerro bat dago token bakoitzeko eta zutabe bat atributu bakoitzeko. CoNLL 2001 biltzarrean hiru atributu soilik erabili ziren; lehena, forma; bigarrena, kateei buruzko informazioa (BIO formatuan); eta azken zutabea, guri interesatzen zaiguna, perpaus-mugak.

Ebaluazio hau aurrera eramateko *train* eta *test* corpusak hautatu ziren. Prozesu hau modu gorakor batean garatu ahal izateko, ataza hiru zatitan banatu zen.

Lortutako emaitzak aztertuta, emaitzarik onenak Carreras (2005) eta Màrquez (2002)ek lortu zituztela ikus daiteke eta honengatik aukeratu da beraiek erabilitako ikasketa automatikorako algoritmoa. Hain zuzen, 91 punturainoko F1 neurria lortu zen testerako erabilitako corpusean.

II.3 *Hitz multzo*en identifikazioa: iragazketa eta sailkapena, *pertzeptroi*ekin

Hizkuntzaren Prozesamenduko zenbait atazatarako beharrezkoa da ikasketa automatikoko sailkatzaileak konbinatzea.

(Carreras, 2005) eta (Carreras *et al.*, 2005) lanetan, zenbait sailkatzaile konbinatzen dituen ikasketa-estrategia orokor bat proposatzen da, *hitz multzo* batzuen izaera errekurtsiboa kontuan izanda (perpausak, kasu). Sistemak, zehatzago esanda, *hitz multzo*en egiturak identifikatzen ditu esaldian, eta bi mailatan edo bi geruzatan lan egiten du:

- Lehenengoan, iragazketa egiten da hitz mailan (*filtering*): esaldiko *hitz multzo* posible guztiak detektatzen dira, hau da, *hitz multzo hautagaiak*. Beste modu batean esanda, hitz bakoitza *hitz multzo* baten hasiera edo bukaera izan daitekeen ala ez erabakitzen da geruza honetan. Auke-

ratutako *hitz multzo* hautagai guztiek ez dute zertan koherente izan esaldiarentzat.

- Bigarrenean, *hitz multzo* mailan lan egiten da. Geruza honetan, lehen geruzan iragazitako *hitz multzo hautagaiak* puntuatzen dira (*ranking*), eta esaldiarentzat *hitz multzo*en segida onena aukeratzen da. Alegia, *hitz multzo* hautagai bakoitzari puntuazio bat ematen zaio —zenbaki erreal bat—, testuinguru horretan *hitz multzo* hori esaldian zenbaterainoko hautagai sendoa den adierazten duena.

Esaldiaren azken puntuazioa, beraz, aukeratutako *hitz multzo* hautagaiek duten puntuazioen batura izango da. *Hitz multzo*ekin aritzeak, ordea, badu desabantaila bat: aztertu beharreko hautagaien konbinazioak asko izan daitezkeela. Hori dela eta, hitz mailan egiten den lanak —hau da, *hitz multzo*en hasierak eta bukaerak aukeratzeak— garrantzi handia du; izan ere, geroz eta *hitz multzo*en hasiera eta bukaera posible gehiago aukeratu, orduan eta *hitz multzo* hautagai gehiago izango ditugu, eta, beraz, baita hautagaien konbinazio posible gehiago ere.

Hortaz, hiru ikasketa-funtzio daude guztira: iragazketako *start* eta *end* funtzioak, hitz bakoitza *hitz multzo* baten hasiera edo bukaera izan ote daitekeen erabaki beharko dutenak hurrenez hurren, eta *score* deiturikoa, *hitz multzo* hautagai bakoitzari puntuazio bat emango diona, hautagaitza horren sendotasunaren arabera.

Iragazketa (*filtering*) eta sailkapena (*ranking*) izeneko bi geruzatan egiten duelako lan eta *pertzeptroi*en algoritmoaren halako orokortze bat baliatzen duenez hiru ikasketa-funtzioak inplementatzeko, *FR-Perceptron* izena jarri zion Carreras-ek (2005) bere algoritmoari. Jarraian xehetasun handiagoz azalduko dugu.

II.3.1 *FR-Perceptron* algoritmoa: iragazketa eta sailkapena, *pertzeptroi*ekin

*Pertzeptroi*en algoritmo tradizionalaren halako orokortze bat da Carreras-en (2005) algoritmoa. *Pertzeptroi*en algoritmoen familiakoa izanik, erroreak gidatutakoa dela esan daiteke. *Hitz multzo*en identifikazio-prozesuan, algoritmoak iteratu egiten du n aldiz; alegia, ikasketa-corpuseko adibide bakoitza n aldiz bisitatzen da (*epoch-zenbakia* deitzen zaio parametro honi). *Start* eta *end* funtzioak esaldiko hitz bakoitzeko aplikatzen dira lehendabizi, eta

score funtzioaren sarrera izango diren *hitz multzo*en hautagaiak definitzen dira honela. Gero, *score* funtzioa aplikatzen zaio, modu errekurtsiboan, *hitz multzo*en hautagai bakoitzari. Honela, *hitz multzo*en konbinazio onena aukeratzen da esaldiko. Egindako iragarpena okerra baldin bada, sailkatzaileak zuzentzen dira hurrengo iteraziorako, erregela sinple batzuen bidez. Esaldiarentzat soluzio onena bilatzen duenez, algoritmo globala dela esaten da (Carreras, 2005).

Ebatzi beharreko problemaren arabera, *hitz multzo*en egitura *jarraituak* edo *errekurtsiboak* bilatuko ditu algoritmoak. Hala, kateen kasuan, *hitz multzo*en egitura *jarraituak* izango ditugu, sintagmak eta aditz-kateak egitura jarraituak baitira; perpausen kasuan, aldiz, *hitz multzo*en egitura *errekurtsiboak* izango ditugu, perpaus bat beste baten baitan joan daitekeelako, hain zuzen ere.

Sailkatzaile guztiak *pertzeptroi*en algoritmoaren hiru aldaerarekin probatu zituen Carreras-ek (2005): *last*, *voted* eta *averaged*. Emaitza onenak *averaged perceptron* (Freund eta Schapire, 1999) delakoak eman zizkion, zeina *pertzeptroi*aren algoritmo klasikoaren hobekuntza sinple bat baita: ikasketa egiterakoan, algoritmo honek zenbait sailkatzaileren konbinazio moduko bat —batez besteko moduko bat— kalkulatzen du. Emaitza onak lortu dira algoritmo honekin HParen alorrean (Collins, 2002). Hala ere, gure esperimentuetan hau konprobatu nahi genuen eta *averaged perceptron*ez gain *last* modua ere erabili genuen.

FR-Perceptron algoritmoarekin, literaturako emaitzarik onentsuenak lortu zituzten, bai kateen identifikazioan (Sang eta Buchholz, 2000), bai perpausen identifikazioan (Sang eta Déjean, 2001). Bi atazotan duen portaera onaz gain, ordea, erabili beharreko algoritmoak beste bi baldintza edo ezinbesteko ezaugarri ere izan behar zituen, ikasketa automatikoko algoritmo gehientsuenek betetzen dituztenak, bestalde:

- Batetik, algoritmoak hizkuntza desberdinetara egokitzeko gaitasuna izan behar zuen, eta horretarako erraztasunak eskaini behar zituen.

- Bestetik, ikasketarako ezaugarri edo atributu berriak gehitzeko aukera eman behar zuen. Ezaugarri hau ezinbestekoa zen guretzat. Izan ere, ingeleserako erabiltzen zen corpusaren tamaina gaztelaniarako erabiltzen genuen corpusaren tamaina berdintsua badu ere, hobekuntzarako tartea izan nahi genuen.

II.4 Perpausen identifikazioa euskararako

Arrieta (2010) tesiaren ekarpen esanguratsuena ikasketa automatikoko teknikak erabiliz euskararen prozesamenduan zenbait aurrerapauso ematea izan zen. Hala, hiru tresnak sortu ziren: euskarako kateen eta perpausen identifikatzaile automatikoak eta koma-zuzentzailea.

Azaleko sintaxiaren baitan, perpaus-identifikatzaile automatiko sendo eta erabilgarri bat sortu zen (F_1 = % 77,24). Ekarpen nagusi honez gain, atal honi dagozkion hauek ere nabarmendu behar dira:

1. *FR-Perceptron* ikasketa-algoritmoa arrakastaz egokitu zen euskararako.

   Algoritmo horrek *hitz multzo*ak identifikatzeko ingelesarekin frogatuta zuen portaera ona berretsi zen, euskarako pareko atazetan ere emaitza onak lortuz.

2. Ezaugarri linguistiko esanguratsuak gehituta hobetu zen perpaus-identifikatzailea.

   Horretarako, ezaugarrien aukeraketa egin behar izan zen, lehenik. Probatu zen informazio linguistiko guztia izan zen baliagarria: hitza, lema, kategoria, azpikategoria, deklinabidea eta mendeko perpausen informazioa.

3. Ikasketa automatikoko teknikak eta hizkuntzaren ezagutzan oinarritutakoak uztartu ziren.

   IXA taldean hizkuntza-ezagutzan oinarritutako tekniken bidez sortutako perpaus-mugatzailea aprobetxatu zen. Horretarako, patroi edo erregela bidez lortutako informazioa txertatu zen ikasketa-algoritmoan, informazio gehigarri gisa (*stacking* edo pilaratzea erabiliz), eta emaitzak hobetu ziren honela. Gainera, hobekuntza hauek estatistikoki esanguratsuak direla frogatu zen.

Tesi honetatik atera daitekeen ondoriorik esanguratsuena euskarazko perpausak identifikatzea ingelesekoak identifikatzea baino zailagoa dela da.

Perpausen identifikazio automatikoari dagokionez, II.1 taulan ikusi daitezke tesi honetan lortutako azken emaitzak, ingelesekoekin erkatuta. Ingeleseko emaitzak baino zazpi puntu gutxiagoko F_1 neurria lortzen da euskararako. Emaitzak, corpus desberdinetan ebaluatuak izanik, ez dira zuzenean

Hizkuntza	**Teknika**	**Desanbiguatua**	F_1
Euskara	*FR-P* oin+ak+d+l+m+Er	Autom	77,24
Ingelesa	*FR-P* oin	Autom	84,36

Taula II.1: Euskarako eta ingeleseko perpaus-identifikatzaileen emaitzen arteko konparaketa (automatikoki analizatutako eta desanbiguatutako corpusa (Autom) eta *FR-Perceptron* (*FR-P*) baliatuta); eta euskarakoaren kasuan, *oinarrizko ezaugarri*ak (oin), azpikategoria (ak), deklinabidea (d), lema (l), mendekoen informazioa (m) eta erregeletan oinarritutako perpausen mugatzaileak emandako informazioa (er) eta ikasketa-corpusaren tamaina osoa (% 100 = 104.956 token) erabilita.

konparagarriak. Hala ere, kasu honetan *oinarrizko neurri*ak bi hizkuntzentzat antzekoak direnez, ingeleserako lortzen den hobekuntza euskararako lortzen dena baino handiagoa da. Honen arrazoi nagusietako bat euskararen ordena librea —edo inguruko hizkuntzena baino libreagoa, behintzat— izan daiteke (Aldezabal *et al.*, 2003). Arrieta (2010) lanean esaten zenez, ingelesez esaldi batean parte hartzen duten elementuen ordena zurrunagoak badu zer esanik emaitzetan, eta alderantziz, euskaraz esaldi bat egiteko orduan hitzen ordena hain finkoa ez izateak ez du laguntzen perpausen identifikazioan, ezen kasuistika zabalago baten aurrean jartzen baitu makina.

II.5 Ondorioak

Ikasketa automatikorako erabiliko dugun programak, datuak zutabeka hartzen dituenez, argi dago Ancoraren corpusak ez digula balio dagoen bezala, bertatik behar ditugun datuak atera beharko dira eta guri egokien datorkigun formatura egokitu, hau da zutabeka jarrita. Ondoren Freeling analizatzaile sintaktikoa erabili beharko dugu corpus berria sortzeko, eta hau berriro ere gure formatura moldatu. Azkenik *FR-Perceptron* algoritmoa erabiltzea erabaki dugu ebaluaketak egiteko, emaitzarik onenak lortu dituen algoritmoa baita CoNLL 2001 batzarreko emaitzek diotenez eta euskararako perpausidentifikatzailea lortzeko arrakastaz moldatu ahal izan delako.

III. KAPITULUA

Proiektuaren helburu dokumentua

III.1 Helburuak

Proiektu hau gaztelaniarako perpaus-identifikatzaile automatiko bat egitean datza, eta berarekin lortutako emaitzak aztertzea.

Horretarako Ancora corpusa CoNLL formatura itzuli beharko da *FR-Perceptron* programak erabili ahal izateko. Gainera, Ancora corpusa osatzen duen jatorrizko testua lortu beharko da, eta Freeling analizatzaile sintaktikoari bidali automatikoki analizatutako corpusa lortzeko. Honi esker emaitzak errealistak izango dira. Automatikoki analizatutako corpusa ere CoNLL formatura moldatu beharko da *FR-Perceptron* programak uler dezan.

Corpusak lortu ondoren, atributuen konbinazio desberdinak erabiliz hainbat proba egin beharko dira sortutako bi corpusekin. Proba hauek, ikasketarako corpusarekin ikasi ostean, garapenerako sortutako corpusekin egingo dira. Behin emaitza guztiak lortuta, test corpusarekin eta emaitza onenak ematen dituen atributuen konbinazioarekin azken probak egingo dira behinbetiko emaitzak lortzeko.

Azkenik lortutako emaitzak konparatu eta interpretatu beharko dira; hala nola, atributuek azken emaitzan duten eragina analizatuko da.

III.2 Lan metodologia

III.2.1 Kudeaketa

Zuzendariaren eta ikaslearen arteko bilerak astean behin egingo dira, printzipioz. Hala ere, egindako lanaren eta aztertu beharrekoaren arabera hurrengo bilera aurreratu edo atzeratuko da, irakaslearen eta ikaslearen artean aurretik adostuta. Bileran eramangarriren baten edukia aztertu behar bada, aurreko egunean irakasleari posta elektronikoko mezu batean bidaliko zaio. Honek, aldi berean, segurtasun-kopia gisa ere balioko digu. Behin bilera burutu ostean, ikasleak bileraren akta bete beharko du, bertan hartu diren erabakiak eta hurrengo bileraren data jarriz.

Segurtasun kopiak 2 astero egingo dira, baina lan-karga handia badago, lehenago egingo da.

Proiektua martxa konstantean eramateko, egunean 3-4 ordu sartzea pentsatu da. Hala ere, proiektuaren gora beherak direla medio, egun batzuetan ordu gehiago sartu beharko dira eta beste batzuetan gutxiago.

Bi pertsonaia nagusi daude proiektuan, zuzendaria eta ikaslea, hau da, proiektuaren garatzailea. Bien arteko komunikazioa posta elektroniko edo telefono bidezkoa izango da, eta bileretan, noski, kontaktu zuzena egongo da.

Bilera hauen aktak eranskinetan aurkitu daitezke.

III.2.2 Baliabideak

Proiektuaren programazio lana Windows makinetan egingo da, Java programazio lengoaian NetBeans programa erabiliz. Hala ere, Freeling eta FR-Perceptron unibertsitateko makinetan instalatuta daude, eta hauek erabiltzeko VPN konexio bat erabiliko da. Programa hauek ikaslearen makinan instalatzea bideragarria ez dela erabaki da, bere konplexutasunagatik eta ekar ditzakeen abantailak ez direlako horrenbesterako.

Segurtasun kopiak MEGAUPLOAD web zerbitzua erabiliz egiten hasi ginen, baina aipatutako web orri honek izandako arazo legalen ondorioz, DROPBOX zerbitzuan egin genituen azken hilabeteetako segurtasun kopiak.

Informazio iturri nagusi gisa Bertol Arrietaren tesia izango dugu, bere tesiaren zati bat proiektu honen oso antzekoa baita. Honez gain, programazio arazoak konpondu nahian internet ere erabili dugu informazio iturri erabilgarri gisa.

III.3 Proiektuaren garapenaren deskribapena

Proiektua garatzean hainbat pauso eman dira: proiektuaren definizioa, aurrekarien azterketa, inplementazioa... Jarraian pauso garrantzitsuenen azalpen labur bat emango da.

Proiektuarekin hasi aurretik, aurretik eginda zegoen lana aztertu behar izan genuen, eta horretarako perpausen identifikazioari buruz ahal genuen guztia irakurri genuen. Honi esker *FR-Perceptron* algoritmoaren nondik norakoak ikasi ahal izan genituen, eta proiektuaren helburua hobeto ulertu.

Proiektuaren helburuak finkatu bezain laster, corpusen formatu-aldaketa programen diseinua egin genuen. Hau garbi izan genuenean, programen inplementazioari ekin genion. Hau egiteko, Java programazio lengoaia erabiliz, XML fitxategietatik datuak atera dira, fitxategia karakterez karaktere tratatuz. Honela hainbat zerrenda sortu dira, eta bakoitzak informazio mota bat gorde du; hala nola, hitzak testuan duen agerpen zehatza, lema, kategoria eta azpikategoria besteak beste. Ondoren, informazio hori testuan aurkitu ahala, listak betetzen joan dira eta behin XML fitxategi guztia tratatu ondoren, listetako informazioa testu fitxategi arrunt batera kopiatzen da, corpusaren gainontzeko XML fitxategiak tratatzen jarraitu ahal izateko.

Azkenean, XML fitxategi guztietatik ateratako informazioa testu-fitxategi bakar batean izango dugu eskuragarri. Testu fitxategi hau izango da, ikasketa automatikoko programari pasa diogun sarrera.

Honekin batera, Ancora corpusetik testu soila ere atera dugu. Behin hau lortuta, Freeling azaleko analizatzaile sintaktikoari bidali zaio. Honek XML fitxategi bat ematen du irteera moduan. XML honi beste bihurketa-prozesu bat aplikatzen zaio ikasketa automatikoko tresnak behar duen formatua izan dezan.

Formatu aldaketa hauek egin ondoren, bi corpusak parekatu dira, Freeling analizatzaileak modu desberdinean interpreta ditzakeen esaldiak kanporatzeko eta lortutako lagina ahalik eta konparagarriena izateko.

Corpusak hiru zati desberdinetan banatu dira, entrenamendurako % 70, garapenerako % 15 eta ebaluaziorako % 15.

Behin beharrezko corpusak lortuta eta egiaztatzeak egin ostean, IXA taldeko zerbitzarietan probak egin dira *FR-Perceptron* programa erabiliz. Horretarako, programaren konfigurazio fitxategiak gure corpusa erabiltzeko gai izan zedin prestatu behar izan ditugu.

Probak amaitutakoan, emaitzak aztertu ditugu eta hainbat hobekuntza

egin ditugu emaitza hobeak lortzeko asmoz. Atributuen aukeraketa landu dugu, batez ere.

Amaitzeko, behar genituen emaitza guztiak esku artean izanda, egindako lan guztia dokumentatu dugu, proiektuan zehar sortutako dokumentuak bertan txertatuz.

III.4 Proiektuaren nondik norakoak

III.4.1 LDE diagrama

Ikus III.1 irudia.

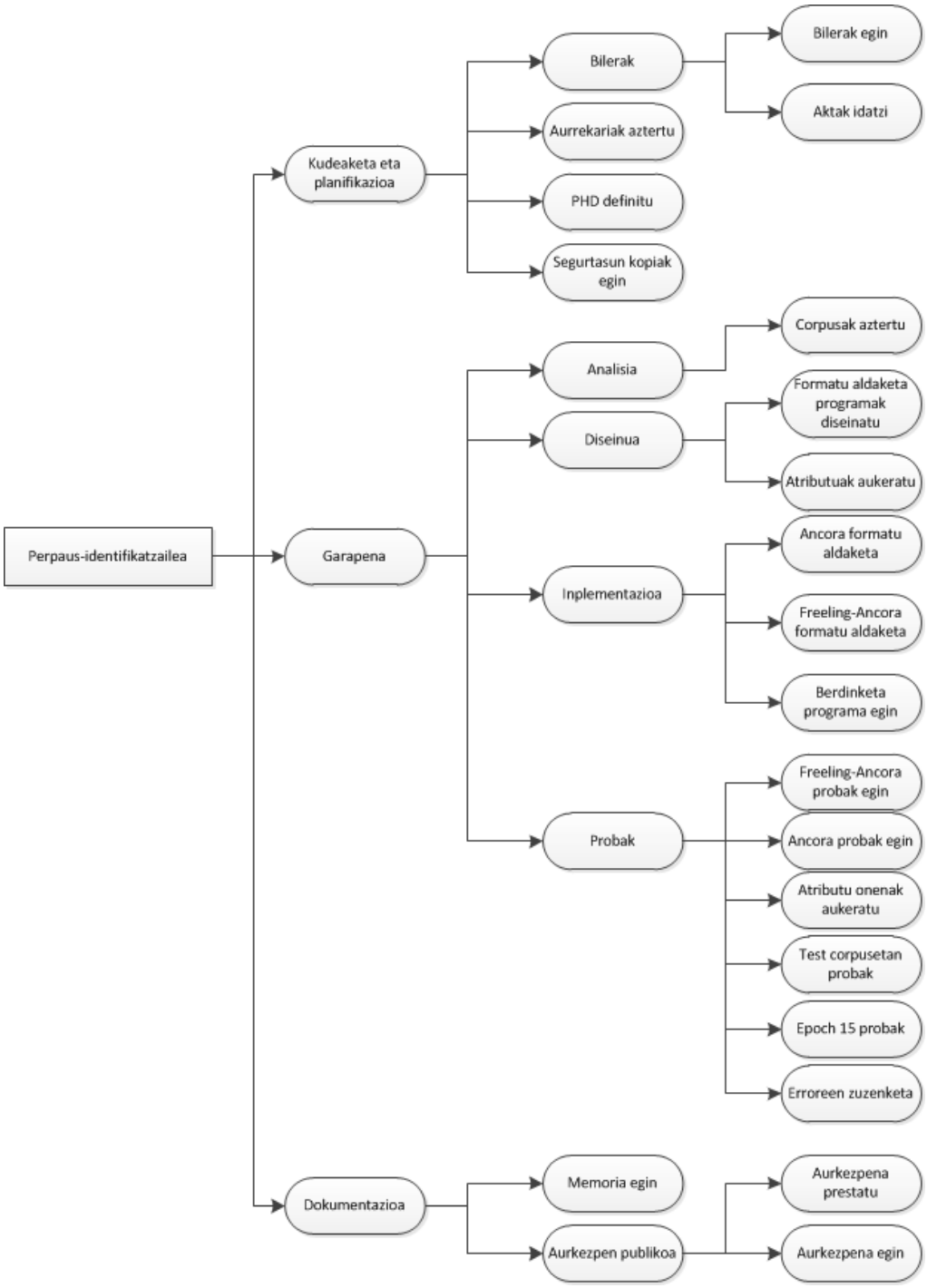
III.4.2 Azpiatazen zerrenda

Prozesu taktikoak:

- Kudeaketa
  - Bilerak
    - Irakasleari eramangarriak bidali
    - Bilera burutu
    - Akta idatzi
    - Hurrengo bilerarako egin beharrekoen zerrenda idatzi
  - Segurtasun kopiak
    - Segurtasun kopiak egin
  - Proiektuaren helburu dokumentua definitu

Prozesu operatiboak:

- Aurrekariak aztertu
  - Bertol Arrietaren tesia irakurri
  - CoNLL 2001 batzarrari buruz irakurri
  - Freelingi buruzko informazioa irakurri
  - Ancora corpusari buruzko informazioa irakurri



Irudia III.1: LDE diagrama

- Garapena
  - Analisia
    - Ancora corpusa aztertu
    - Freeling-Ancora corpusa aztertu
  - Diseinua
    - Ancora formatu aldaketa programa diseinatu
    - Freeling-Ancora formatu aldaketa programa diseinatu
      - Perpaus-mugen informazioa gehitzeko programa diseinatu
    - Bi corpusak berdintzeko programa diseinatu
  - Inplementazioa
    - Ancora formatu aldaketa programa inplementatu
    - Freeling-Ancora formatu aldaketa programa inplementatu
      - Perpaus-mugen informazioa gehitzeko programa inplementatu
    - Bi corpusak berdintzeko programa inplementatu
    - Corpusak sortu
    - Konprobaketak egin
  - Probak
    - Garapen corpusarekin probak
      - Freeling-Ancora corpusarekin probak
      - Ancora corpusarekin probak
      - Atributu konbinazio onena aukeratu
      - Erroreen zuzenketa
      - Epoch 15 erabiliz probak egin
    - Test corpusarekin behin-betiko emaitzak lortu
      - Freeling-Ancora test corpusean atributu onenekin proba egin
      - Ancora test corpusean atributu onenekin proba egin
- Dokumentazioa
  - Memoria egin

- Aurkezpen publikoa
  - Aurkezpen publikoa prestatu
  - Aurkezpen publikoa egin

III.5 Planifikazioa

Estimatutako ahalegina eta ahalegin errealaren arteko konparaketa taula III.2 irudian ikusi dezakegu.

III.2 irudian ikus daitekeen moduan, hasierako estimazioen arabera proiektuaren garapen osoak 313 ordu inguru emango zituela estimatu genuen. Estimatu genuen inplementazioa izango zela denbora gehien emango zigun ataza, beti ere, inplementazio garaian sortzen baitira espero ez diren arazoak eta hauek konpontzeak denbora asko eman dezake kasu batzuetan. Bestalde, diseinu aldetik, ez genuen denbora gehiegi estimatu, egin beharreko programek ez baitzuten diseinu aldetik konplexutasun handirik, nahiz eta modu egokian inplementatzeak denbora eraman digun.

III.3 grafikan ikus daitekeen bezala, kasu gehienetan estimatutako orduak gaindituak izan dira eskainitako denbora errealarekin. Inplementazioaren kasuan, sortutako arazoak izan dira honen arrazoia; dokumentazioan, aldiz, egin beharreko zuzenketek baldintzatu dute ordu gehiago sartu behar izana.

Salbuespen gisa, aurrekarien azterketan ikus dezakegu, non estimatutako orduak baino gutxiago behar izan ditugun. Egin beharreko irakurketak hasiera batean uste genuena baino azkarrago egin ziren.

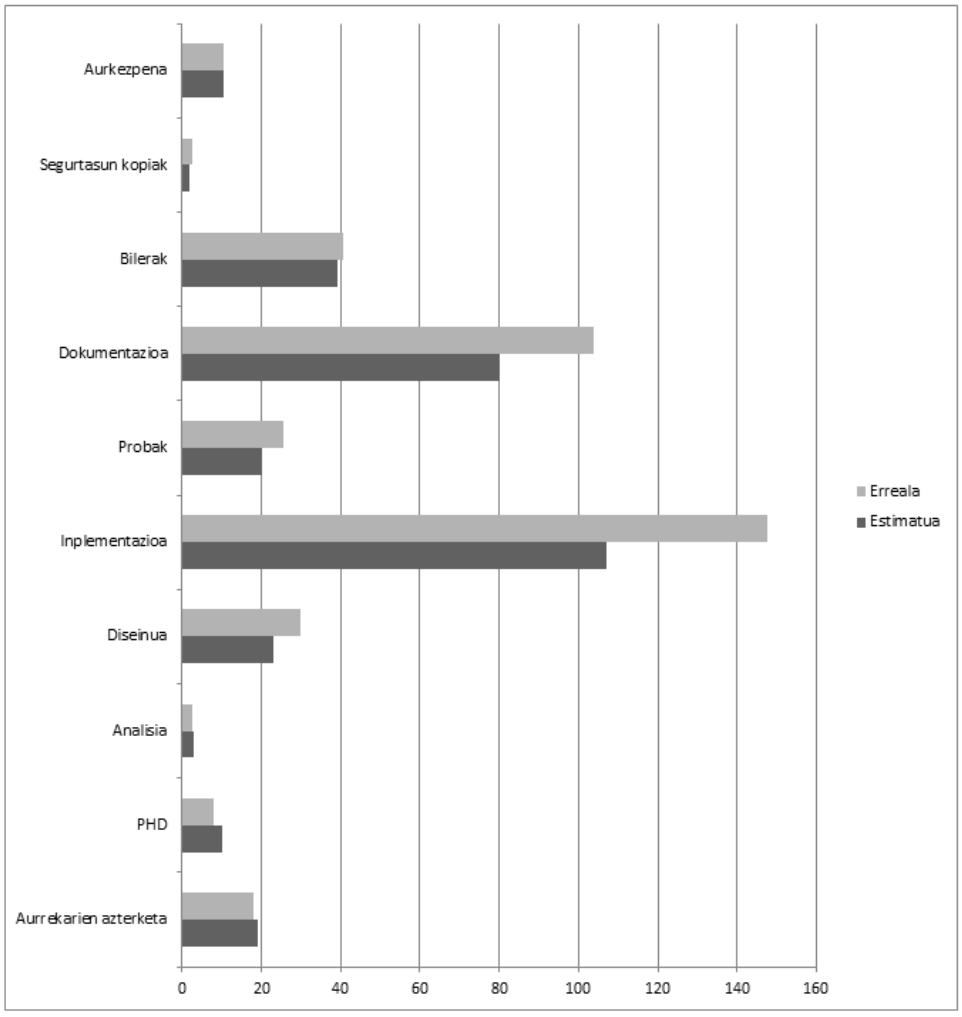
Egindako estimazioen arabera, astean 5 egun lan egingo dira, egunero 3 ordu inguru sartuz. Hau, ordea, hurbilpen soila da, egin beharreko lan kopuruaren lan-kargaren arabera egun batzuetan 6 ordu ere sartu behar izan dira.

III.4 irudiko plangintza honen arabera, maiatzaren amaierarako proiektuak amaitua beharko luke. Errealitatean, ordea, III.5 irudian ikus daitekeen bezala lana luzatu egin zaigu ekainaren 2. astera arte.

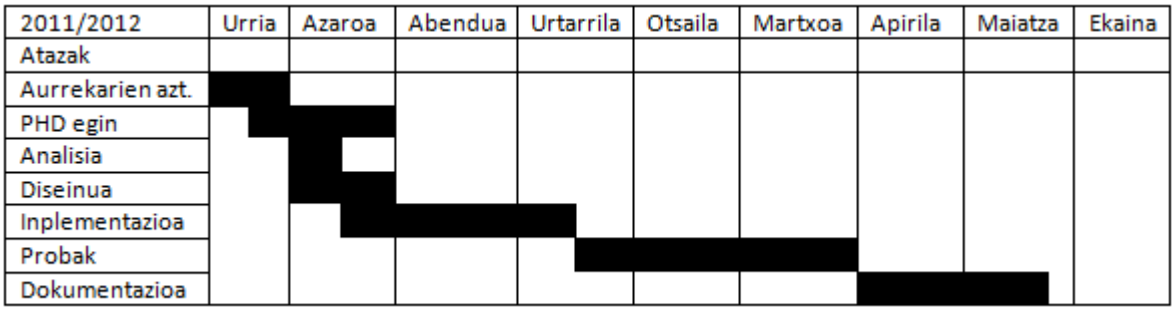
Taulan honetan ikus daitekeen desberdintasun nagusia, inplementazioaren zatian da. Hasiera batean, programazio lan guztia gabonetarako amaitzea pentsatua baigenuen. Freeling-Ancora corpusaren moldaketekin programazio lana bikoiztu egin zitzaigun gabonetan. Ancora corpusarekin egindako programak bere horretan ezin izan genituelako erabili. Gainera, bi corpus parekatzeak beti ematen du lana, eta uste baino denbora gehiago behar izan

	ESTIMAZIOAK	ERREALA
GUZTIRA	313:30	388:40
Prozesu taktikoak	51:00	51:15
Kudeaketa	51:00	51:15
Bilerak	39:00	40:45
Irakasleari eramangarriak bidali	2:00	2:00
Bilera burutu	26:00	32:00
Akta idatzi	7:00	3:00
Hurrengo bilerarako egin beharrekoen zerrenda idatzi	4:00	3:45
Segurtasun kopiak	2:00	2:30
Segurtasun kopiak egin	2:00	2:30
Proiektuaren helburu dokumentua definitu	10:00	8:00
Prozesu operatiboak	262:30	337:25
Aurrekariak aztertu	19:00	17:50
Bertol Arrietaren tesia irakurri	16:00	12:30
Ancora web orriko dokumentazioa irakurri	3:00	5:20
Garapena	153:00	205:15
Analisia	3:00	2:30
Ancora corpusa aztertu	2:00	1:00
Freeling-Ancora corpusa aztertu	1:00	1:30
Diseinua	23:00	29:45
Ancora formatu alaketa programa diseinatu	15:00	14:45
Freeling-Ancora formatu aldaketa programa diseinatu	7:00	13:30
Perpaus-mugen informazioa gehitzeko programa diseinatu	1:00	2:00
Bi corpusak berdintzeko programa diseinatu	1:00	1:30
Inplementazioa	107:00	147:30
Ancora formatu aldaketa programa inplementatu	43:00	61:00
Freeling-Ancora formatu aldaketa programa inplementatu	40:00	59:00
Perpaus-mugen informazioa gehitzeko programa inplement.	20:00	30:00
Bi corpusak berdintzeko programa inplementatu	10:00	15:00
Corpusak sortu	12:00	10:30
Konprobaketak egin	2:00	2:00
Probak	20:00	25:30
Garapen corpusarekin probak	18:00	23:30
Freeling-Ancora corpusarekin probak	2:00	6:00
Ancora corpusarekin probak	6:00	8:30
Atributu konbinazio onena aukeratu	2:00	1:30
Epoch 15 probak	2:00	3:00
Erroreen zuzenketa	6:00	4:30
Test corpusarekin behin-betiko emaitzak lortu	2:00	2:00
Freeling-Ancora test corpusean atributu onenekin proba egin	1:00	1:00
Ancora test corpusean atributu onenekin proba egin	1:00	1:00
Dokumentazioa	80:00	103:50
Memoria egin	80:00	103:50
Aurkezpen publikoa	10:30	10:30
Aurkezpen publikoa prestatu	10:00	10:00
Aurkezpen publikoa egin	00:30	00:30

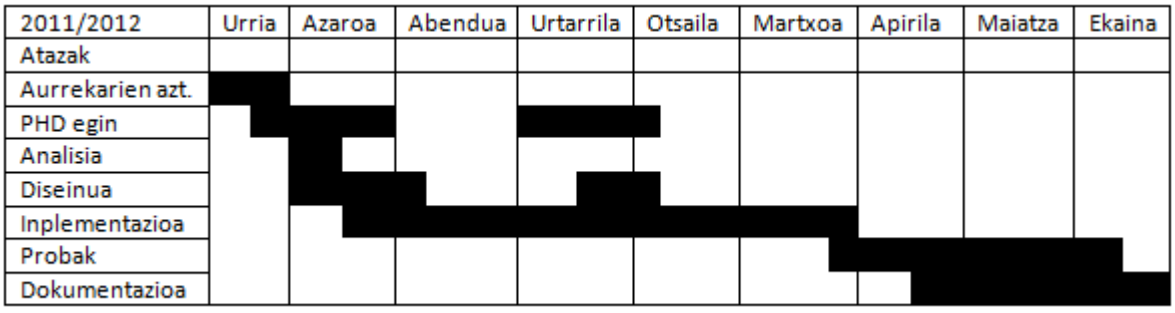
Irudia III.2: Estimatutako ordu kopurua eta ordu kopuru errealen taula



Irudia III.3: Estimatutako orduen eta ordu errealen grafika



Irudia III.4: Gantt diagrama estimatua



Irudia III.5: Gantt diagrama erreala

genuen. Honen ondorioz, urtarrilean hastea pentsatzen genituen probak martxoraino atzeratu behar izan genituen, corpus guztiak prest izan arte. Honek, noski, dena atzeratu digu, baita dokumentazioaren eta proiektuaren entregadata ere.

III.6 Arriskuak

III.6.1 Arriskuen plana

- Materiala hondatzea: Informazio galera posibleari interneten gordetako segurtasun kopiekin aurre egingo zaio. Segurtasun kopiak 2 astero egingo dira, azken eguneraketak ongi babestuak izateko. Honez gain, bilerak egin aurretik, bileretan erabiliko den materiala posta elektronikoz bidaliko zaio irakasleari, berarekin adostu ondoren, fitxategi hauek ere segurtasun kopia gisa edukitzeko. Proiektua garatzean 2 konputagailu erabiliko dira, beraz askotan bi ordenagailuetan egongo dira lanaren kopiak. Interneten gordetzeko erabilitako web orria bertan behera geratuz gero, beste web zerbitzu bat bilatu beharko da eta bertan egin segurtasun kopia berriak.

- FR-Perceptron eta Freelingekin egon daitezkeen arazoak: Erabiliko ditugun programa hauek arazoren bat emanez gero, beraien egileekin kontaktuan jarri beharko dugu eta arazoa konpontzen saiatu ahalik eta denbora gutxien galtzeko.

- Birusek sortutako informazio galerak: Ez luke arazo handirik suposatu behar, segurtasun kopia eguneratuak baliatuz proiektua garatzen jarraitu daiteke. Hala ere, beharrezkoak diren segurtasun-neurriak erabiliko dira lana egingo den Windows makinetan: antibirus eguneratua eta suhesia, batik bat.

- Ezusteko arazo pertsonalak: Ezusteko arazo pertsonalen bat izanez gero, gaixo jartzea adibidez, galdutako denbora berreskuratu beharko da asteburuetan edo aste tartean planifikatua baino ordu gehiago sartuz.

- Lanean gauden bitartean argi-indarra joatea: Gaur egungo Word bertsioak egoera horiek saihesteko segurtasun-kopia sistema bat du, eta dokumentuen azken bertsioa berreskuratzeko oso eraginkorra da. Hala

ere, portatilean lan egitean argi indarraren momentuko mozketak ez luke arazo hau sortuko.

III.6.2 Arriskuen kontrola

Goian aipatutako arriskuak kontrolatzeko, erabilitako ordenagailuak ahalik eta eguneratuen edukitzen saiatu gara. Besteak beste, antibirusa eguneratuta edukitzen saiatu gara, baita windowsen segurtasun eguneraketak aldiro instalatzen.

Proiektuaren arrisku nagusiena eta kontrolatzen zailena egin zaiguna ezusteko arazo pertsonalak izan dira. Hauek, lana aurreratzen ekidin daitezke, baina ezustean gaixo jarriz gero, hor galdutako orduan aurrerago berreskuratu beharko dira, eta horrek askotan denbora galtzea suposatzen du.

Segurtasun kopiak egitean *Megaupload* zerbitzua erabiltzeak arazoak sortu dizkigu. Jakina denez, zerbitzu famatu hau arazo legalak direla medio, zerbitzuz kanpo geratu zen. Honen ondorioz, segurtasun kopiak egiteko zerbitzuz aldatu beharrean aurkitu ginen eta *Dropbox* zerbitzua erabiltzea erabaki genuen. Aurrerago ikusi dugu, zerbitzu hau askoz erabilgarriagoa eta erosoagoa izan daitekeela segurtasun kopiak egiteko.

IV. KAPITULUA

Garapen Teknikoa

Kapitulu honetan, proiektua aurrera eramateko erabili diren elementuak eta egindako proben nondik norakoak azalduko dira.

IV.1 Esperimentuen prestaketa

Atal honetan, erabili ditugun corpusak deskribatuko ditugu. Bestalde, ebaluatzeko zein neurri erabili ditugun ere azalduko dugu, eta bukatzeko, esku artean darabiltzagun atazetarako lortutako oinarrizko neurriak aurkeztuko ditugu. FR-Perceptron algoritmoak esaldi osorako soluzio global bat proposatzen du, eta, beraz, esaldiko hitz guztiak hartzen dira kontuan soluzio onena bilatzeko. Hala ere, leihoen tamainaren arabera hitzaren ondorengo eta aurreko hitzen informazioa kontuan hartzen du hitz mailako erabakiak hartzeko. Gure esperimentuetan (-3,+3) leihoa erabili dugu. Ez zaigu interesgarria iruditu lehio tamaina desberdinekin probak egitea, FR-Perceptron algoritmoak esaldi osoa hartzen baitu kontuan.

IV.1.1 Ancora corpusa

Ancora corpusak 500.000 token inguru ditu, eta neurri handi batean egunkarietako testuez osatua dago. Corpusa maila askotan etiketatua izan da, baina ingeleseko perpaus-identifikatzailearekin konparatu nahi dugunez, konparazioa fidelagoa izatearren, ingeleserako erabilitako informazio linguistikoa

soilik erabili dugu hasieran. Hala ere, interesgarria iruditu zaigun informazio guztia baliatu dugu beste hainbat probatan, informazio gehiago izateak ekarriko lituzkeen abaintailak ebaluatzeko. Ancora corpusa FR-Perceptronekin erabili ahal izateko formatu berezi batera bihurtu behar izan dugu, formatu hau 2001. urteko CoNLL batzarreko ataza partekatuan erabilitako bera izango da, eta FR-Perceptronek sarrera gisa erabiliko du. Formatu honi, hemendik haurrera *CoNLL formatua* deituko diogu.

Ancora jatorrizko corpusa hainbat XML fitxategitan banatua dago, osatzen duten testuen informazio-iturrien arabera eta notazio mailaren sakontasunaren arabera. Guk, ordea, fitxategi guztiak berdin tratatu ditugu, eta behar genuen informazioa CoNLL formatura egokitu. Hala nola, tokenen forma, lema, kategoria eta azpikategoria zuzenean XML fitxategiko elementu bakoitzaren atributuetatik atera ditugu. Honez gain, perpausaren mugen informazioa eta token bakoitzaren katearen informazioa atera dugu bi modu ezberdinetan errepresentatua jarriz. Bata, BIO formatuan, hau da, token bakoitzean kate bat hasten den (B), tokena aurreko kate baten barruan dagoen (I) ala katerik ez duen(O), jarraian kate motaren izena izanik.

- B-GV: *group-verb* motako katea hasten da.
- I-SN: *sn* motako katearen barruan dago.
- O: kateetatik kanpo dago.

Honez gain, kateak beste modu batera errepresentatzea ere erabaki dugu, askotan, BIO formatuan informazioa galdu baitaiteke, adibidez, kateen amaierak ondoriozta daitezkeen arren, askotan ezin da jakin non amaitu den kate bakoitza. Beraz, parentesiez baliatuz token bakoitzean hasten diren eta amaitzen diren kate guztiak azaltzea erabaki genuen, modu honetara:

(SN*SN)SP): etiketa honek adieraziko luke izen-sintagma bat hasi, beste bat bukatu eta preposizio sintagma bat bukatzen dela.

Informazio guztia atera ondoren, CoNLL formatuan utzi behar izan dugu corpusa ondoko itxurarekin:

Adibidea IV.1.1

Desde	*desde*	*s*	*preposition*	*B-SP*	*(SP**	*(S**
su	*su*	*d*	*possessive*	*B-SN*	*(SN**	**
rehabilitación	*rehabilitación*	*n*	*common*	*I-SN*	**SN)*	**
,	*,*	*f*	*comma*	*I-SP*	**SP)*	**

Ikasketa automatikoa egiterakoan, ikasi nahi ditugun perpaus-mugak eskuz etiketatuak izan behar duten gisan, ikasteko darabilgun gainerako informazio linguistikoa komeni da analizatzaile automatiko baten bidez lortutakoa izatea, emaitza errealistak lortu nahi badira. Kontuan hartu behar baita, hain zuzen ere, ikasi duguna, testu berri baten gainean aplikatzerakoan, testu berri horren analisi automatikoa soilik erabili ahal izango dugula, eta ez aditu batek eskuz etiketatua edo zuzendua.

Ikasketa-prozesuan erabilitako informazio linguistikoa automatikoki lortua izatea emaitzen kalterako izango da, noski, geroz eta informazio linguistiko hobea, orduan eta ikasketa-eredu hobea lortzen baita. Baina, esan bezala, emaitzarik errealistenak lortzea dugu helburu, eta hori lortuko dugu modu honetan lan eginda. Esan nahi baita automatikoki lortutako informazioa baliatuko dela edozein aplikazio praktiko egiteko, eta informazio hau erabiltzen duen jokalekua dela, honenbestez, egoera errealena.

Hala ere, ikasketa-prozesuan eskuzko informazio linguistikoa erabiltzeak gure ikasketa-ereduaren ahalmena erakusten digu, hau da, automatikoki lortutako informazio linguistikoa erabat zuzena izatera hurbiltzen bada, lortutako emaitzak noraino irits daitezkeen.

Hau guztia kontuan izanda, esan dezakegu gure probetarako bi corpus desberdin erabili ditugula, Ancora corpusa eta Freeling-Ancora deitu dioguna. Biak testu berdinak, baina informazio linguistikoa modu desberdinean lortua dutenak, lehenak eskuz etiketatua eta bigarrenak automatikoki lortua. Perpaus-mugen informazio bera izango dute bi corpusek: eskuz etiketatua.

IV.1.2 Freeling-Ancora corpusa

Freeling-Ancora corpusa deitu dioguna Ancora corpusa osatzen duen jatorrizko testutik abiatuta lortutakoa da. Horretarako, jatorrizko testua Freeling analizatzaile sintaktikoari pasa diogu, eta honek irteera bezala XML fitxategi batean jatorrizko testua hainbat informazio linguistikorekin itzuli digu. Hau egin ahal izateko, lehendabizi, Ancora corpusetik testu soila lortu behar izan genuen, ez baikenuen corpusa osatzen zuen jatorrizko testua lortzerik izan.

Behin Ancora testua Freelingek analizatuta, XML fitxategitik behar genuen informazio linguistikoa atera genuen, hau da, tokenaren forma, lema, kategoria eta katearen informazioa. Katearen informazioa lortzean ohartu ginen Ancora corpusarekin zegoen desberdintasun nabarmen batekin. Freeling programa beraren izaeragatik, puntuazio ikurrak kate mota baten barruan sartzen dira. Ancora corpusean berriz O bezala identifikatu dira.

Hona hemen Freeling-Ancora corpusaren zati baten adibide bat:

Adibidea IV.1.2

Una	*uno*	*DI*	*B-sn*
portavoz	*portavoz*	*NC*	*I-sn*
de	*de*	*SP*	*B-sp-de*

Hau Freeling analizatzaile sintaktikoak emandako informazioa formatuz aldatu ondoren daukagun corpusa da. Beste formatu-aldaketa prozesu batetik pasa behar da, token bakoitzaren lerro amaieran perpaus-mugen informazioa gehitu behar baitugu.

IV.1.3 Corpusaren moldaketa

Ancora jatorrizko corpusa CoNLL formatura pasa ondoren, corpusa hiru zati desberdinetan banatu dugu: train (% 70), develop (% 15) eta test (% 15). Train zatia FR-Perceptron ikasketa automatikoko programa entrenatzeko erabili dugu, eta develop, entrenamendu horren emaitzak probatzen joateko, atributuen konbinazio desberdinekin. Test zatia atributuen konbinazio optimoa lortu ondoren erabiliko dugu, azken emaitza lortzeko. Hiru zati hauek, ordea, behin-behinekoak dira, Freeling analizatzaile sintaktikoak Ancora corpusa osatzen duen jatorrizko testua analizatu ostean, corpus hauetan moldaketa batzuk egin behar izan baitira, jarraian azalduko dugun moduan.

Formatu-aldaketa prozesuetan hainbat esaldi kanpoan utzi beharrean aurkitu gara. Freeling-Ancora eta Ancora corpusen arteko konparaketa on bat egiteko asmoz, Freelingek Ancora corpusa analizatzean kakotxak zituzten esaldiekin arazoak zituen, batzuetan esaldi baten barruko aipamen bat baitzen, eta Freelingek beste esaldi bat bezala interpretatzen zuen. Ikus dezagun ondorengo adibidea hau, garbiago ulertzeko:
"Hay que subir el IVA", dijo el presidente del gobierno.
Freelingek esaldi hau analizatzean bi esaldi bereiziko lituzke, Ancorako corpusean esaldi bakar baten gisan agertzen den bitartean:
"Hay que subir el IVA" eta *dijo el presidente del gobierno.*
Honen ondorioz esaldi hauek kanpoan utzi behar izan ditugu eta hasieran genituen ia 500.000 tokenetatik 340.000 inguru token dituen corpusarekin geratu gara. Honez gain, Freeling-Ancora corpusa sortzean, kontuan izan behar da azken zutabean perpaus-mugen informazioa gehitu behar zaiola, beraz esaldi bakoitzak duen tokenen segidak zehazki berdina izan behar du

corpus batean eta bestean, bestela perpaus-informazioa txertatzean gerta liteke bat ez etortzea. Horren ondorioz, formatu-aldaketa programetan, aldiro konprobaketa ugari egin dira, eta bat ez zetozen esaldiak alde batera utzi behar izan dira.

Corpusak ahalik eta konparagarrien izateko asmoz, esaldi berdinez osatuak izatea bermatu nahi izan dugu. Horretarako, formatu-aldaketa prozesuetan hainbat konprobaketa egin dira, Freeling analizatzaileak testua analizatzean interpretazio desberdinak egiten baititu. Freelingen konfigurazio fitxategia ahalik eta egokien jarri ondoren, ordea, esaldi batzuk kolokan geratu zaizkigu, ez baitzetozen bat Ancora jatorrizko corpusan agertzen zenarekin. Datak eta portzentaiak jartzerakoan, Freelingek token bat erabili beharrean bi edo hiru token erabiltzen zituen, eta honek arazoak sortzen zituen perpaus-mugen informazioa txertatzean. Kasu hauetan, beraz, esaldi hauek kanpoan utzi behar izan ditugu.

Prozesu hau guztia ahalik eta automatizatuen egiten saiatu garen arren, eskuz konponketa batzuk egin behar izan ditugu Freeling-Ancora corpusean. Freeling analizatzailearen konfigurazioak eta Ancora jatorrizko corpusaren notazio-arazoak eraginda, ezinezkoa izan zaigu Freeling analizatzailearentzat konfigurazio perfektu bat aurkitzea, arazo bat konpontzean beste bat sortzen baitzen. Arazoak sortzen zituzten kasuak, 'del' eta 'al' moduko hitzak ziren, Freelingek analizatzean 'de+el' eta 'a+el' bezala erazagutzen baitzituen, eta honi eskuz etiketatutako perpaus-mugen informazioa gehitzean arazoak sortzen zitzaizkigun, lerro bat beharrean bi baitzeuden. Beraz, eraginkorren iruditu zaigun konfigurazioa utzi dugu eta gaizki geratutako esaldiak eskuz konpondu behar izan ditugu (hutsuneak sartuz elkartuta agertzen ziren bi esaldi desberdin banatzeko, gehienbat).

Ondorengo taulan ikus daiteke gure corpusen banaketa. Kontuan izan behar da CoNLL batzarreko esperimentuetan 260.000 token inguruko corpusa erabili zutela; zehatz esanda, 211.727 token ikasteko, eta beste 47.377 token probarako. Beraz, datuok ikusita, esan daiteke emaitzak konparagarriak direla corpusaren tamainaren aldetik.

	Token kopurua	**train**	**develop**	**test**
Freeling-Ancora	316.515	220.168	48.221	48.126
CoNLL	259.104	211.727	-	47.377

Taula IV.1: Perpausen identifikaziorako erabilitako corpusaren neurria.

Kontuan izan behar da, corpusaren banaketa egiteko garaian 7-2-1 eta 7-1-2 banaketa erabili dela. Hau da, corpusa analizatzean, lehenengo iterazioko 7 lehenengo esaldiak train corpusera eraman dira, hurrengo biak develop corpusera eta hurrengoa test corpusera. Hurrengo iterazioan, lehenengo 7 esaldiak train corpusera joan dira, hurrengo esaldia developera, eta hurrengo biak berriz test corpusera. Honi esker % 70-% 15% 15 banaketa egitea lortu da.

IV.1.4 CoNLL formatua

Behin eta berriz aipatzen ari gara CoNLL formatua. CoNLL formatua 2000 eta 2001 urteetako ataza partekatuetan erabili zen (kateen eta perpausen identifikazioan, hurrenez hurren) eta FR-Perceptron algoritmoa erabili ahal izateko formatu honetara moldatu behar izan dugu Ancora corpusa.

CoNLL formatuan ikasketarako zein probarako corpusek lerroz eta zutabez osatutako matrize antzeko bat osatzen dute. Lerro bakoitzean token bat izango dugu jatorrizko testuaren ordenan; hutsune batez banaturiko zutabeetan, berriz, atributuak izango ditugul. Azken zutabean ikasi nahi dugunari buruzko informazio zuzena (eskuz etiketatua) izango dugu. Perpausen identifikaziorako, esaterako, hauek erabili ziren CoNLL 2001 batzarrean: hitzaren forma, kategoria, hitz-katearen informazioari buruzko etiketa (katea) eta ikasi beharreko balioa, perpaus-mugen informazioa. Lehen aipatu bezala, azken-bigarren zutabe honek 3 balio izan ditzake:

- B-KATE, katearen hasiera adierazteko (adibidez, B-NP, (*begin noun-phrase*), sintagmaren hasiera adierazteko).
- I-KATE, katearen barnean dagoela adierazteko (adibidez, I-VP (*in verb-phrase*), aditz-kate barneko parte dela adierazteko).
- O , kateetatik kanpo dagoela adierazteko.

Azken zutabea da gure kasuan garrantzitsuena, ikasi beharreko balioa baitu. Perpaus-muga etiketak honako balio hauek edo hauen konbinazioak izan ditzake("*" zeinua, konbinazio bakoitzean, behin bakarrik agertuko da):

- (S*: perpausaren hasiera adierazteko.
- *S): perpausaren amaiera adierazteko.

- *: tokena ez da perpausaren hasiera, ezta perpausaren bukaera ere.

Hala nola, (S(S*S) etiketak bi perpausen hasiera dela esan nahi du, eta aldi berean, beste perpaus baten bukaera dela.

IV.1.5 Ebaluaziorako neurriak

Ebaluazioari dagokionez, analizatzaile sintaktiko automatikoak ebaluatzeko PARSEVAL neurri hauek (Black *et al.*, 1991) erabili ohi izan dira: doitasuna (*precision*) eta estaldura (*recall*). Hartutako erabakien zuzentasuna neurtzen du doitasunak; hau da, jarritako perpaus-muga ondo jarrita dagoen ala ez. Estaldurak, berriz, zuzenak direnetatik asmatzen direnen portzentaia ematen du, beste modu batera esanda, jarri behar liratekeen perpaus-mugetatik zenbat jarri diren. Hizkuntzaren Prozesamenduan eskuz etiketatutako osagaiak hartzen dira zuzentzat.

Neurri hauek guztiak IV.2 gisako kontingentzia-taula batean oinarrituz kalkulatzen dira, bi klaseko emaitzak (Y eta N) ditugunean.

	Zuzena=Y	**Zuzena=N**
Esleitua=Y	a	b
Esleitua=N	c	d

Taula IV.2: Estatistikak kalkulatzeko kontingentzia-taula.

"a" zenbakiak Y klasekoak diren eta Y klasea esleitu zaien elementuen kopurua adierazten du; "b" zenbakiak N klasekoak diren baina Y klasea esleitu zaien elementuen kopurua; "c" zenbakiak Y klasekoak diren baina N klasea esleitu zaien elementuen kopurua; eta "d" zenbakiak, berriz, N klasekoak diren eta N klasea esleitu zaien elementuen kopurua. Gure kasuan, Y klasea, perpaus-muga jartzea izango litzteke, eta N klasea, perpaus-muga ez jartzea.

Doitasuna eta estaldura emaitzaren klase posible bakoitzeko kalkulatzen dira. Oro har, honela definitzen dira bi neurri hauek Hizkuntzaren Prozesamenduan, analizatzaileak neurtzen gabiltzanean (A_z analizatzaile automatikoak *zuzen* etiketatutako osagai kopurua izanik; A_e analizatzaile automatikoak *etiketatutako* osagai kopurua izanik; E_e *eskuz* etiketatutako osagai kopurua (zuzentzat hartutakoak) izanik):

$$Doitasuna = A_z/A_e$$
$$Estaldura = A_z/E_e$$

IV.2 taulako kontingentzia-taula kontuan harturik, Y klaseko datuak, esaterako, honela kalkulatuko lirateke:

$$Doitasuna = a/(a+b)$$
$$Estaldura = a/(a+c)$$

Doitasunaren eta estalduraren artean erlazio matematiko zuzenik ez dagoen arren, estudio enpirikoetan ikusi ahal izan denez, alderantziz erlazionatuta daude; alegia, sistemak detektatutako elementuen kopurua handitzen bada —hots, estaldura handitzen bada—, doitasuna txikitzen da, eta alderantziz. Ondorioz, bi neurri hauek batera konparatzea ez da erraza. Horregatik, bi neurriok kontuan hartzen dituzten zenbait neurri proposatu izan dira. Gehien erabiltzen dena $F_{\mathcal{B}}$ neurria da.

$$F_{\mathcal{B}} = \frac{(\mathcal{B}^2+1)*Doitasuna*Estaldura}{(\mathcal{B}^2*Doitasuna+Estaldura)}$$

Normalean, $\mathcal{B} = 1$ erabiltzen denes, guk ere hau erabiliko dugu, doitasunari eta estaldurari pisu bera emanez:

$$F_1 = \frac{2*Doitasuna*Estaldura}{(Doitasuna+Estaldura)}$$

Kasu batzuetan, zehaztasuna edo *accuracy* izeneko neurria ere erabiltzen da: hartutako erabaki guztietatik zuzenak izan direnen portzentaia neurtzen du. Kontingentzia-taulako datuekin kalkulatzen da neurri hau ere (ikus IV.2 taula):

$$zehaztasuna = (a+d)/(a+b+c+d)$$

Neurri hau, ordea, zenbait kasutan ez da nahikoa esanguratsua; izan ere, garrantzi bera ematen dio emaitza-klase bati edo besteari. Ataza batzuetan, ordea —klase bateko balio askoz gehiago ditugunetan, batez ere— normala izaten da askotan gertatzen den klaserako emaitza onak lortzea eta txarrak, berriz, gutxitan gertatzen den klaserako. Zehaztasunak emaitza-klaseak kontuan hartzen ez dituenez, aipatutako kasuetan ez da neurri esanguratsua izaten. Hori dela eta, klase bakoitzarekiko kalkulatzen den F_1 neurria erabiltzen da, oro har, eta hala egin dugu guk geuk ere.

Perpausen identifikazioan, beraz, honako hau adierazten dute:

- Doitasuna: automatikoki detektatuko perpausetatik zenbat diren zuzenak.

- Estaldura: detektatu beharreko perpausetatik (zuzenetatik, alegia) zenbat detektatu diren automatikoki.

Neurri hauek berak erabili ziren ingeleseko kateen eta perpausen identifikazioko atazetan (Sang eta Déjean, 2001; Sang eta Buchholz, 2000) eta baita euskararako perpausen eta kateen identifikazioan ere (Arrieta, 2010).

IV.1.6 Oinarrizko neurriak

Hizkuntzaren Prozesamenduan, abiapuntu bat izan ohi da, hortik aurrera emaitzak hobetzen joateko proba ezberdinak egiten diren heinean. Abiapuntu honi *baseline* edo *oinarrizko neurri* deitzen zaio. Azpiatal honetan, perpausen identifikaziorako aukeratu dugun oinarrizko neurria azalduko dugu, eta hau nola lortu den.

Perpausen identifikazioan hartutako habiapuntua CoNLL 2001eko batzarrean erabilitako heuristikoa izan da: esaldiaren hasierako eta bukaerako tokenei soilik jartzen zaie perpaus-muga, esaldia puntutik puntura doan unitate gisa hartuta. CoNLL 2001eko batzarrean lortutako F_1 neurria % 47,71koa izan zen, Euskarazko perpaus-identifikatzailearena, berriz, (Arrieta, 2010) % 48,79koa izan zen. Argi ikusten da, beraz, heuristiko honekin hizkuntza batetik bestera ez dela aldaketarik, heuristikoaren izaeran bertan baitago hau honela izatearen zergatia: doitasun handia lortzen da(% 98,44), baina estaldura txikia(% 31,48). Gaztelaniarako kasuan heuristiko hau erabiliz lortutako F_1 neurria % 44,46 izan da.

IV.2 Diseinua

Corpusak CoNLL formatura pasatzeko eta beharrezko moldaketak egiteko hainbat programa egin behar izan genituen eta hauen funtzionamendua azalduko dugu. Baita ere, *FR-Perceptron* tresnan eginiko moldaketak aipatuko ditugu.

IV.2.1 Formatu aldaketarako programak

Egin behar izan genuen lehenengo programa, Ancora jatorrizko corpusa guk behar genuen formatura pasatzeko izan zen. Ancora corpusa, hainbat XML fitxategiz osatua dago, eta gure programak hauek guztiak banan banan tratatu behar zituen eta bertatik guk behar genuen informazioa atera.

Lehendabiziko programak, sarrera gisa guk ezarritako direktorioan zeuden XML fitxategi guztiak hartzen zituen, eta irteera gisa, corpusaren 3 fitxategi ematen zituen txt formatuan; train.txt, develop.txt eta test.txt.

XML fitxategietatik informazioa ateratzeko, XML eredu horren egitura aztertu behar izan genuen eta behin interesatzen zitzaizkigun atributuak identifikatuta hauek nola inprimatu erabaki genuen. Informazio gehiena XML atributuak ziren, baina kateen mugak identifikatzea konplexuagoa izan zen. Hautatu genuen soluzioa, pila datu-egitura bat erabiltzea izan zen. Bertan, sintagma mota bat hasten zenean honen identifikatzailea pilaratzen genuen, eta sintagma amaitzen zenean despilaratu egiten genuen. Hau oso erabilgarria izan zitzaigun katearen informazio gehigarria modu erraz batean inprimatzeko, eta emaitzetan ikusiko den bezala, informazio hau oso garrantzitsua izan zaigu emaitzak hobetzeko.

Bestalde, Ancora corpusa osatzen zuen jatorrizko testua lortu ahal izan genuen, aurrerago Freeling analizatzaile sintaktikoari bidaltzeko.

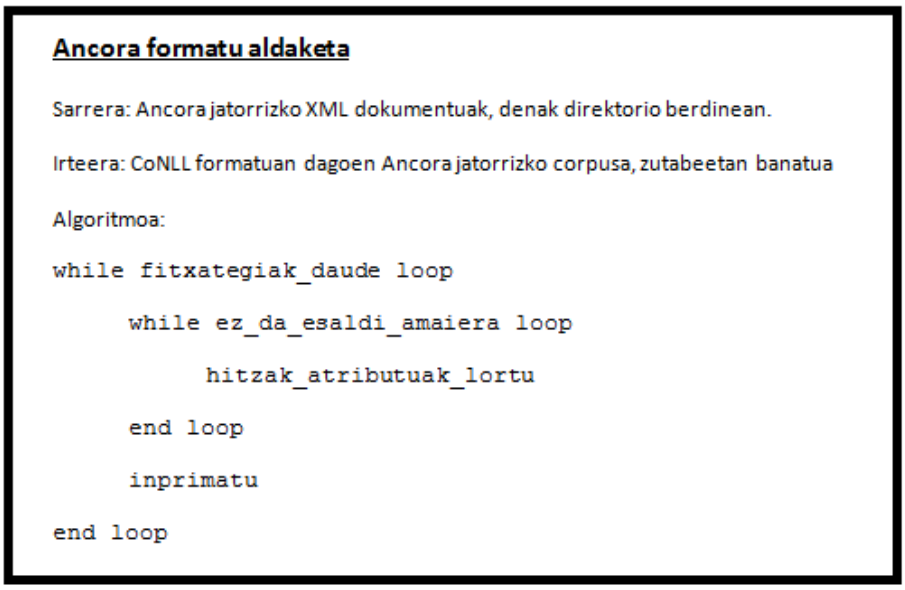
IV.1 eta IV.2 irudietan argiago ikus ditzakegu programa honen nondik norakoak.

Behin Ancora corpusa gure formatura bihurtuta, Freeling-Ancora corpusa sortzeko garaia zen.

Freeling azaleko analizatzaile sintaktikoak, sarrera gisa txt formatuan dagoen testu-fitxategi bat erabiltzen du, eta irteera gisa, XML formatuan dagoen fitxategi bakar bat ematen du. XML fitxategi honetatik behar genuen informazioa ateratzeko ez zigun balio Ancora corpuserako eginiko programak, XML fitxategia modu guztiz desberdin batean eratua baitzegoen.

Bigarren programa honek sarrera gisa XML fitxategia du, eta irteera gisa txt moduan, gure formatura bihurtutako fitxategi bat, IV.3 irudian ikus daitekeen moduan.

Behin bi corpusak edukita, Freeling-Ancora corpusari, ikasketa automatikoko programak ikasi beharreko informazioa gehitzea falta zitzaigun, hau da, Ancora corpuseko azken zutabea, perpaus-mugen informazioa duena. Informazio hau txertatu ahal izateko, hirugarren programa bat egin behar izan genuen. Hirugarren programa honek sarrera gisa Ancora corpusaren txt fi-



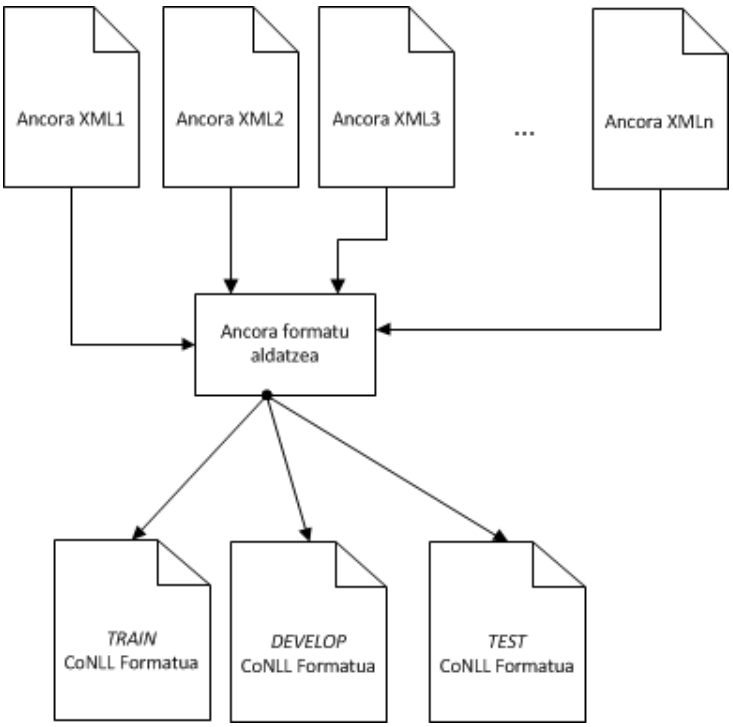
Irudia IV.1: Ancora jatorrizko testua CoNLL formatura itzultzeko programa

txategia (train.txt, develop.txt ala test.txt) eta honen baliokide zen Freeling-Ancora txt fitxategia hartzen zituen, eta, irteera gisa, Freeling-Ancora txt fitxategi bat ematen zuen, baina oraingoan perpaus-mugen informazioa lerro bakoitzaren amaieran duelarik.

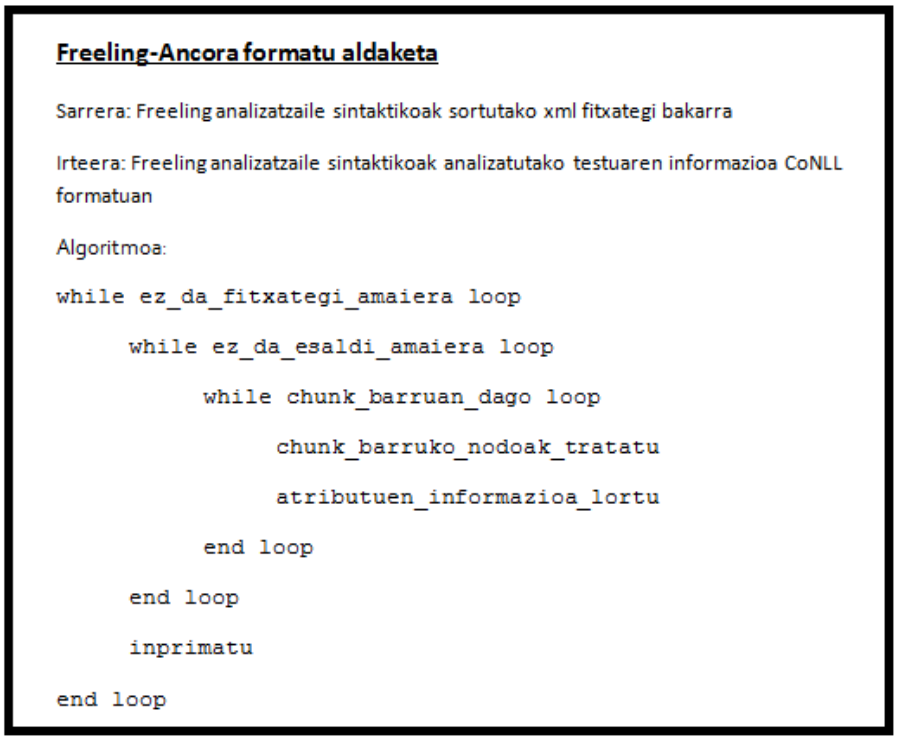
Programa honek sarrera fitxategiak paraleloan pasatzen ditu, eta bi esaldien artean (Ancora fitxategian eta Freeling-Ancora fitxategian) desberdintasun bat badago, esaldi hau kanporatua geratzen da, eta ez da irteera fitxategian inprimatuko. Honi esker, Freeling analizatzaileak modu desberdin batean interpretatutako esaldiek ez digute arazorik sortzen eta bi corpusak berdinak izatea bermatzen dugu. Ikus IV.4 irudia.

Honekin amaitzean, behin-betiko Freeling-Ancora corpusa genuen esku artean, baina Ancora corpusean zeuden esaldi batzuek kanpoan geratu behar izan zuten zehazki lerro kopuru berdina ez zutelako. Honen ondorioz, laugarren programa bat egin behar izan genuen, Ancora corpusetik, Freeling-Ancora corpusean ez zeuden esaldiak kentzeko. Azken programa honek irteera bezala trainberdindua.txt, developberdindua.txt eta testberdindua.txt fitxategiak emango ditu.

Aipatutako programez gain, zenbait programa txiki ere egin dira konpro-



Irudia IV.2: Ancora jatorrizko testua CoNLL formatura itzultzeko programaren beste ikuspegi bat



Freeling-Ancora formatu aldaketa

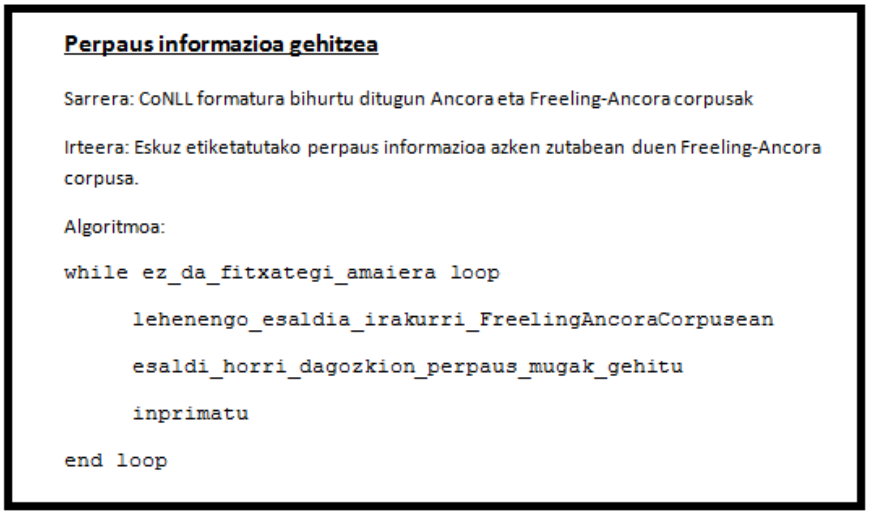
Sarrera: Freeling analizatzaile sintaktikoak sortutako xml fitxategi bakarra

Irteera: Freeling analizatzaile sintaktikoak analizatutako testuaren informazioa CoNLL formatuan

Algoritmoa:

```
while ez_da_fitxategi_amaiera loop
      while ez_da_esaldi_amaiera loop
            while chunk_barruan_dago loop
                  chunk_barruko_nodoak_tratatu
                  atributuen_informazioa_lortu
            end loop
      end loop
      inprimatu
end loop
```

Irudia IV.3: Freeling analizatzaileak sortutako XML fitxategia CoNLL formatura itzultzeko programa



Irudia IV.4: Freeling-Ancora corpusari perpaus informazioa gehitzeko programa

baketa batzuk egitea komenigarria iruditu zitzaigulako. Honi esker, azken orduko arazoak ekidin ditugu, informazioa modu egokian hartzen ari gineneko segurtasuna baikenuen. Adibidez, Ancora corpusean informazioa ongi hartzen ari ginela ziurtatzeko, laugarren eta bostgarren zutabeek (azpikategoria eta katearen informazioa BIO formatuan) har zitzazkeen balioen zerrendak ateratzeko azpiprograma bat egin genuen. Zerrenda hauek oso luzeak ez zirenez eskuz begiratuta ikus zitekeen balio arrarorik ez zegoela zerrendatuta zeuden hitz kateen artean.

IV.2.2 FR-Perceptron moldatzen

FR-Perceptron programa gure corpusekin erabili ahal izateko, hainbat fitxategi aldatu behar izan genituen; zehazki, *clausefex.pm, sentence.pm* eta *word.pm*. Fitxategi hauetan zenbait aldaketa egin genituen atributu linguistiko berriak definitu eta *FR-Perceptron*ek kontuan har zitzan. Aldi berean, bi corpus desberdin erabili behar genituenez, fitxategi bakoitzaren bi bertsio sortu behar izan genituen, bata Ancora corpusarentzat eta bestea Freeling-Ancora corpusarentzat. Beti ere, jatorrizko fitxategien segurtasun kopia bat

mantendu genuen direktorio berean, zerbait gaizki ateraz gero arazorik gabe konpondu ahal izateko.

Ondorengo irudietan *clausefex.pm, sentence.pm* eta *word.pm* konfigurazio fitxategietan Ancora eta Freeling-Ancora corpusak tratatu ahal izateko eginiko aldaketak ikus daitezke, ez dira konfigurazio fitxategi osoak, aldatutako zatiak bakarrik. Fitxategiak osorik eranskinetan gehitu ditugu.

```
#@{$fex->{countWords}} = ( "non","zein","zeina","ez", "eta");
#previous line ALSO commented by Andoni Ibirriaga 2012/03/26 and changed with next line:
@{$fex->{countWords}} = ( "que");
#@{$fex->{countPOS}} = ( "''", "``", "(", ")", ",", ".", ":", "?","!",";","...");
#previous line ALSO commented by Andoni Ibirriaga 2012/03/26 and changed with next line:
@{$fex->{countPOS}} = ( "''", "``", "(", ")", ",", ".", ":");
#@{$fex->{countChunks}} = ("DET");
#previous previous line ALSO commented by Andoni Ibirriaga 2012/03/26 and changed with next line:
@{$fex->{countChunks}} = ( "c", "p");
# c para los "que" subordinados y "p" para los "cuyo,quien..." relativos
-----------------------------------------------------------------------------------------------
#if ($w->form eq "that") {
#$item = "that";
#}
#previous if commented by Andoni Ibirriaga and changed by:
if ($w->form eq "que") {
   $item = "que";}
#elsif ($w->pos =~ /^W/) {
#$item = "W";
#}
#elsif ($c->type =~ /^V/) {
#$item = "V";
#}
#previous 2 elsif commented by Andoni Ibirriaga and changed by the 2 following ones:
 elsif ($w->pos =~ /^c/) {
  $item = "c";}
 elsif ($w->pos =~ /^p/) {
  $item = "p";}
-----------------------------------------------------------------------------------------------
  if ($w->pos =~ /^(''|``|\(|\)|,|\.|:|CC)/) {
      #$item = $w->pos;
      }
      #elsif ($w->form eq "that") {
      #$item = "that";
      #}
      #previous elsif commented by Andoni and changed by the next elsif:
      elsif ($w->form eq "que") {
      $item = "que";}
      #elsif ($w->pos =~ /^W/) {
      #$item = "W";
      #}
      #previous elsif commented by Andoni and changed by the 2 next elsif
      elsif ($w->pos =~ /^c/) {
      $item = "c";}
      elsif ($w->pos =~ /^p/) {
      $item = "p";}
-----------------------------------------------------------------------------------------------
      push @F, $label.":$k:pos:".$W[$j]->pos;
      if ($W[$j]->pos =~ /$reWords/) {
      push @F, $label.":$k:form:".lc($W[$j]->form);}
      #Andoni Ibirriaga:
      if ($W[$j]->pos =~ /$reWords/) {
      push @F, $label.":$k:lemma:".lc($W[$j]->lemma);}
      if ($W[$j]->pos =~ /$reWords/) {
      push @F, $label.":$k:subcat:".$W[$j]->subcat;}
      if ($W[$j]->pos =~ /$reWords/) {
      push @F, $label.":$k:chunkcomplex:".$W[$j]->chunkcomplex;}
```

Irudia IV.5: *clausefex.pm* fitxategiari eginiko aldaketak Ancora corpusa tratatzeko gai izateko.

```
my $char;
my $pos = 0;
my $form_field = -1;
my $pos_field = -1;
my $chunk_field = -1;
my $phrase_field = -1;
#Andoni Ibirriaga:
my $lemma_field = -1;
my $subcat_field = -1;
my $chunkcomplex_field = -1;

foreach $char ( split('', $input_spec) ) {
if ( $char eq 'w') {
    $form_field = $pos;
}
elsif ( $char eq 'p') {
    $pos_field = $pos;
}
elsif ( $char eq 'c') {
    $chunk_field = $pos;
    $chunk_tagging = "IOB2";
}
elsif ( $char eq 'C') {
    $chunk_field = $pos;
    $chunk_tagging = "IOB1";
}
elsif ( $char eq 's') {
    $phrase_field = $pos;
    $phrase_tagging = "SE";
}
#Andoni Ibirriaga modif.:
elsif ( $char eq 'l') {
    $lemma_field = $pos;
}
elsif ( $char eq 'b') {
    $subcat_field = $pos;
}
elsif ( $char eq 'd') {
    $chunkcomplex_field = $pos;
}
#end Andoni Ibirriaga modif.

PHRECO::word::set_form_field($form_field);
PHRECO::word::set_pos_field($pos_field);
PHRECO::word::set_chunk_field($chunk_field);
PHRECO::word::set_phrase_field($phrase_field);
#Andoni Ibirriaga modif.:
PHRECO::word::set_lemma_field($lemma_field);
PHRECO::word::set_subcat_field($subcat_field);
PHRECO::word::set_chunkcomplex_field($chunkcomplex_field);
 #end Andoni Ibirriaga modif.:
```

Irudia IV.6: *sentence.pm* fitxategiari eginiko aldaketak Ancora corpusa tratatzeko gai izateko.

```
my $form_field = 0;
my $pos_field = 1;
my $chunk_field = 2;
my $phrase_field = 3; #perp
#Andoni Ibirriaga, gaztelaniako perp-ident:
my $lemma_field = 4;
my $subcat_field = 5;
my $chunkcomplex_field=6;

sub set_form_field {
    $form_field = shift;}
sub set_pos_field {
    $pos_field = shift;}
sub set_chunk_field {
    $chunk_field = shift;
}
sub set_phrase_field {
    $phrase_field = shift;}
#Andoni Ibirriaga, gaztelaniako perp-ident:
sub set_lemma_field {
    $lemma_field = shift;}
sub set_subcat_field {
    $subcat_field = shift;}
sub set_chunkcomplex_field {
    $chunkcomplex_field = shift;
--------------------------------------------------------------------------------
sub new {
  my ($pkg, $id, @fields) = @_;

  my $w = [];
  $w->[0] = $id;

  # form
  $w->[1] = ( $form_field >= 0 ) ? $fields[$form_field] : undef;
  # pos
  $w->[2] = ( $pos_field >= 0 ) ? $fields[$pos_field] : undef;
  # chunk tag
  $w->[3] = ( $chunk_field >= 0 ) ? $fields[$chunk_field] : undef;
  # phrase tag
  $w->[4] = ( $phrase_field >= 0 ) ? $fields[$phrase_field] : undef;

  # Andoni Ibirriaga, gaztelaniako perp-ident:
  #: to add lemma, subcat, chunkcomplex
  # lemma
  $w->[5] = ( $lemma_field >= 0 ) ? $fields[$lemma_field] : undef;
  # subcategory
  $w->[6] = ( $subcat_field >= 0 ) ? $fields[$subcat_field] : undef;
  # chunk complex
  $w->[7] = ( $chunkcomplex_field >= 0 ) ? $fields[$chunkcomplex_field] : undef;
  #end modif.
  #Andoni Ibirriaga modif. of indexes in array after adding
  #lemma, subcat, decl and subord attributes:
  @{$w->[8]} = @fields;
  $w->[9] = undef;
  @{$w->[10]} = ();
  return bless $w, $pkg;
}
```

Irudia IV.7: *word.pm* fitxategiari eginiko aldaketak Ancora corpusa tratatzeko gai izateko 1/2.

```
sub id {
    my $w = shift;
    return $w->[0];}
sub form {
    my $w = shift;
    return $w->[1];}
sub pos {
    my $w = shift;
    return $w->[2];}
sub chunk_tag {
    my $w = shift;
    return $w->[3];}
sub phrase_tag {
    my $w = shift;
    return $w->[4];}
# Andoni Ibirriaga adds these (lemma, subcat, decl, subord)
sub lemma {
    my $w = shift;
    return $w->[5];}
sub subcat {
    my $w = shift;
    return $w->[6];}
sub chunkcomplex {
    my $w = shift;
    return $w->[7];}
------------------------------------------------------------------
# Andoni Ibirriaga modif. these (5->8, 6->9. 7->10)
sub field {
    my ($w, $f) = @_;
    return $w->[8][$f];}
sub fields {
    my ($w) = @_;
    return @{$w->[8]};}
sub set_input {
    my $w = shift;
    $w->[9] = shift;}
sub input {
    my $w = shift;
    return $w->[9];}
sub predictions {
    my $w = shift;
    return @{$w->[10]};}
sub push_predictions {
    my $w = shift;
    push @{$w->[10]}, @_;}
sub set_predictions {
    my $w = shift;
    @{$w->[10]} = @_;}
```

Irudia IV.8: *word.pm* fitxategiari eginiko aldaketak Ancora corpusa tratatzeko gai izateko 2/2.

```
#@{$fex->{countWords}} = ( "non","zein","zeina","ez", "eta");
#previous line ALSO commented by Andoni Ibirriaga 2012/03/26 and changed with next line:
@{$fex->{countWords}} = ( "que");
#@{$fex->{countPOS}} = ( "''", "``", "(", ")", ",", ".", ":", "?","!",";","...");
#previous line ALSO commented by Andoni Ibirriaga 2012/03/26 and changed with next line:
@{$fex->{countPOS}} = ( "''", "``", "(", ")", ",", ".", ":");
#@{$fex->{countChunks}} = ("DET");
#previous previous line ALSO commented by Andoni Ibirriaga 2012/03/26 and changed with next line:
@{$fex->{countChunks}} = ( "C", "P");
# c para los "que" subordinados y "p" para los "cuyo,quien..." relativos
```

```
  #if ($w->form eq "that") {
  #$item = "that";
  #}
  #previous if commented by Andoni Ibirriaga and changed by:
  if ($w->form eq "que") {
  $item = "que";}
  #elsif ($w->pos =~ /^W/) {
  #$item = "W";
  #}
  #elsif ($c->type =~ /^V/) {
  #$item = "V";
  #}
  #previous 2 elsif commented by Andoni Ibirriaga and changed by the 2 following ones:
  elsif ($w->pos =~ /^c/) {
  $item = "c";}
  elsif ($w->pos =~ /^p/) {
  $item = "p";}
```

```
  if ($w->pos =~ /^(''|``|\(|\),|\.|:|CC)/) {
      #$item = $w->pos;}
      #elsif ($w->form eq "that") {
      #$item = "that";}
      #previous elsif commented by Andoni and changed by the next elsif:
      elsif ($w->form eq "que") {
      $item = "que";}
      #elsif ($w->pos =~ /^W/) {
      #$item = "W";}
      #previous elsif commented by Andoni and changed by the 2 next elsif
      elsif ($w->pos =~ /^C/) {
      $item = "C";}
      elsif ($w->pos =~ /^P/) {
      $item = "P";}
```

```
      push @F, $label.":$k:pos:".$W[$j]->pos;
      if ($W[$j]->pos =~ /$reWords/) {
      push @F, $label.":$k:form:".lc($W[$j]->form);}
      #Andoni Ibirriaga:
      if ($W[$j]->pos =~ /$reWords/) {
      push @F, $label.":$k:lemma:".lc($W[$j]->lemma);}
      #if ($W[$j]->pos =~ /$reWords/) {
      #   push @F, $label.":$k:subcat:".$W[$j]->subcat;}
      #if ($W[$j]->pos =~ /$reWords/) {
      #   push @F, $label.":$k:chunkcomplex:".$W[$j]->chunkcomplex;}
      #end modif. Andoni Ibirriaga
```

Irudia IV.9: *clausefex.pm* fitxategiari eginiko aldaketak Freeling-Ancora corpusa tratatzeko gai izateko.

```
my $char;
my $pos = 0;
my $form_field = -1;
my $pos_field = -1;
my $chunk_field = -1;
my $phrase_field = -1;
#Andoni Ibirriaga:
my $lemma_field = -1;
#my $subcat_field = -1;
#my $chunkcomplex_field = -1;

foreach $char ( split('', $input_spec) ) {
if ( $char eq 'w') {
    $form_field = $pos;
}
elsif ( $char eq 'p') {
    $pos_field = $pos;
}
elsif ( $char eq 'c') {
    $chunk_field = $pos;
    $chunk_tagging = "IOB2";
}
elsif ( $char eq 'C') {
    $chunk_field = $pos;
    $chunk_tagging = "IOB1";
}
elsif ( $char eq 's') {
    $phrase_field = $pos;
    $phrase_tagging = "SE";
}
#Andoni Ibirriaga modif.:
elsif ( $char eq 'l') {
    $lemma_field = $pos;
}
#elsif ( $char eq 'b') {
#    $subcat_field = $pos;
#}
#elsif ( $char eq 'd') {
#    $chunkcomplex_field = $pos;
#}
#end Andoni Ibirriaga modif.

PHRECO::word::set_form_field($form_field);
PHRECO::word::set_pos_field($pos_field);
PHRECO::word::set_chunk_field($chunk_field);
PHRECO::word::set_phrase_field($phrase_field);
#Andoni Ibirriaga modif.:
PHRECO::word::set_lemma_field($lemma_field);
#PHRECO::word::set_subcat_field($subcat_field);
#PHRECO::word::set_chunkcomplex_field($chunkcomplex_field);
 #end Andoni Ibirriaga modif.:
```

Irudia IV.10: *sentence.pm* fitxategiari eginiko aldaketak Freeling-Ancora corpusa tratatzeko gai izateko.

```
my $form_field = 0;
my $pos_field = 1;
my $chunk_field = 2;
my $phrase_field = 3; #perp
#Andoni Ibirriaga, gaztelaniako perp-ident:
my $lemma_field = 4;
#my $subcat_field = 5;
#my $chunkcomplex_field=6;
#my $decl_field = 6;
#my $subord_field = 7;
#my $chunk_mg_field = 8;

sub set_form_field {
    $form_field = shift;}
sub set_pos_field {
    $pos_field = shift;}
sub set_chunk_field {
    $chunk_field = shift;}
sub set_phrase_field {
    $phrase_field = shift;}
#Andoni Ibirriaga, gaztelaniako perp-ident:
sub set_lemma_field {
    $lemma_field = shift;}
#sub set_subcat_field {
#    $subcat_field = shift;
#}
#sub set_chunkcomplex_field {
#    $chunkcomplex_field = shift;
#}
sub new {
  my ($pkg, $id, @fields) = @_;
  my $w = [];
  $w->[0] = $id;
  # form
  $w->[1] = ( $form_field >= 0 ) ? $fields[$form_field] : undef;
  # pos
  $w->[2] = ( $pos_field >= 0 ) ? $fields[$pos_field] : undef;
  # chunk tag
  $w->[3] = ( $chunk_field >= 0 ) ? $fields[$chunk_field] : undef;
  # phrase tag
  $w->[4] = ( $phrase_field >= 0 ) ? $fields[$phrase_field] : undef;
  # Andoni Ibirriaga, gaztelaniako perp-ident: : to add
  #lemma, subcat, chunkcomplex
  # lemma
  $w->[5] = ( $lemma_field >= 0 ) ? $fields[$lemma_field] : undef;
  # subcategory
  #$w->[6] = ( $subcat_field >= 0 ) ? $fields[$subcat_field] : undef;
  # chunk complex
  #$w->[7] = ( $chunkcomplex_field >= 0 ) ? $fields[$chunkcomplex_field] : undef;
  #end modif.
  #Andoni Ibirriaga modif. of indexes in array after adding
  #lemma, subcat, decl and subord attributes:
  @{$w->[6]} = @fields;
  $w->[7] = undef;
  @{$w->[8]} = ();
  return bless $w, $pkg;
}
```

Irudia IV.11: *word.pm* fitxategiari eginiko aldaketak Freeling-Ancora corpusa tratatzeko gai izateko 1/2.

```
# id, form, pos, chunk_tag, phrase_tag
sub id {
    my $w = shift;
    return $w->[0];}
sub form {
    my $w = shift;
    return $w->[1];}
sub pos {
    my $w = shift;
    return $w->[2];}
sub chunk_tag {
    my $w = shift;
    return $w->[3];}
sub phrase_tag {
    my $w = shift;
    return $w->[4];}
# Andoni Ibirriaga adds these (lemma, subcat, decl, subord)
sub lemma {
    my $w = shift;
    return $w->[5];}
#sub subcat {
#    my $w = shift;
#    return $w->[6];}
#sub chunkcomplex {
#    my $w = shift;
#    return $w->[7];}

# Andoni Ibirriaga modif. these
sub field {
    my ($w, $f) = @_;
    return $w->[6][$f];
}

sub fields {
    my ($w) = @_;
    return @{$w->[6]};}
sub set_input {
    my $w = shift;
    $w->[7] = shift;}
sub input {
    my $w = shift;
    return $w->[7];}
sub predictions {
    my $w = shift;
    return @{$w->[8]};}
sub push_predictions {
    my $w = shift;
    push @{$w->[8]}, @_;}
sub set_predictions {
    my $w = shift;
    @{$w->[8]} = @_;}
```

Irudia IV.12: *word.pm* fitxategiari eginiko aldaketak Freeling-Ancora corpusa tratatzeko gai izateko 2/2.

IV.3 Perpausen identifikazio automatikoa

Perpausen identifikazioa kateenarekin konparatzen bada ere, badu konplexutasun gehigarria; izan ere, perpausek definizioz izaera errekurtsiboa dute. Honek esan nahi du perpaus bat beste baten barruan egon daitekeela; adibidez: *Tengo una casa que es muy grande*. Adibide honetan esaldi osoa bera perpaus bat izango litzateke, baina gainera, *que es muy grande* beste perpaus bat izango litzateke, aurrekoaren barruan dagoena.

IV.3.1 Teknologiaren aukeraketa

Perpausak identifikatzeko, ikasketa automatikoko eta pertzeptroiekin garatutako *iragazketa eta sailkapena* teknika erabiltzea erabaki genuen. Izan ere, algoritmo hau hitz multzoen identifikazioaz baliatzen da, edozein delarik hitz multzo horren izaera: kateak, perpausak... Arrazoi hau funtsezkoa izan da algoritmo honen aldeko erabakia hartzeko. Gainera, emaitza onak eman ditu, bai ingeleseko testuetan, baita euskarazko testuetan perpausak identifikatzean.

Pertzeptroiekin garatutako iragazketa eta sailkapena erabili ahal izateko, lehen aipatu bezala, CoNLL formatura moldatu behar izan genuen Ancora corpusa eta Freelingek ematen zigun informazioa. Gure kasuan erabilitako *epoch-zenbakia* 10 izan zen hasiera batean, baina aurrerago 15 zenbakiarekin probatu dugu emaitzak hobetzen ziren ikusteko.

IV.3.2 Lehen probak, Freeling-Ancora corpusean

Lehenengo probetan, CoNLL 2001eko batzarrean baliatu zen informazio linguistiko bera erabili genuen: forma, kategoria eta katea. Hau da, *lema* zutabea izan ezik beste zutabe guztiak. Ezaugarri hauei (CoNLL formatukoak: forma, kategoria, katea) oinarrizko ezaugarriak deitu diegu.

Probak egiteko, *FR-Perceptron* tresnak eskaintzen zituen bi modu desberdin erabiltzea erabaki genuen: *last* eta *average*, nahiz eta jakin *AVG* dela emaitza onenak ematen dituena. Aurrekarietan azaldu den bezala, *averaged perceptron* algoritmoa, *pertzeptroi*aren algoritmo klasikoaren hobekuntza sinple bat da: ikasketa egiterakoan, algoritmo honek zenbait sailkatzaileren konbinazio moduko bat kalkulatzen du. Hona hemen oinarrizko ezaugarriekin eginiko lehen probak.

	Doit.	**Est.**	F_1
Oinarrizko ezaugarriekin AVG	85,39	60,84	71,05
Oinarrizko ezaugarriekin LAST	78,98	60,41	68,45

Taula IV.3: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak, Freeling-Ancora corpuseko oinarrizko ezaugarriak erabilita eta 10eko *epoch-zenbakia*. Garapen-corpusaren gainean egindako ebaluazioa.

Emaitzotan *AVG* moduan eginiko ebaluazioak LAST moduan egindakoak baino emaitza hobeak ematen ditu. Esan beharra dago, ebaluazio guztietan gertatuko den zerbait dela, eta desberdintasuna gehienbat doitasunean dagoela ohartu gara.

Bestalde, aitortu beharra dago, ebaluazio honetan ikusitako emaitzak nahiko txarrak iruditu zaizkigula. Izan ere, ingeleseko perpaus-identifikatzaileak, corpus tamaina berdinarekin eta atributu berdinekin, % 84,36eko emaitzak baititu. Euskararen kasuan esaldi barruko elementuen ordena aldagarriak justifikatu lezake oinarrizko ezaugarri hauekin % 69,99ko F_1 neurria bakarrik lortzea. Baina iruditzen zaigu, ingelesez ez litzatekeela gaztelaniaz baino askoz errazagoa izan behar perpausak identifikatzea.

Emaitza txar hauen zergatia aurkitu nahian, hurrengo probak eskuz etiketatutako corpusarekin egin genituen, nolabait, irits gintezkeen emaitzen muga lehenbaitlehen ezagutzeko.

IV.3.3 Ancora corpusean, emaitzen mugen bila

Esan bezala, Freeling-Ancora corpusarekin lor genitzakeen emaitzen hurbilpen bat lortzeko asmoarekin, Ancora corpusa erabiltzea erabaki genuen proba gehiago egiten saiatu aurretik. Horretarako, gainera, Ancora corpuserako atera genituen atributu guztiekin egin genuen proba, baina emaitzak ez ziren guk esperotakoak izan, IV.4 taulan ikusi daitekeen moduan.

Freeling-Ancora corpusarekin lortutako emaitzak baino txarragoak direla kontuan izanik corpus hau eskuz etiketatua dagoela eta informazio gehiago dugula atributuei esker, emaitzak nahiko harrigarriak dira. Arrazoi posibleen artean egon daiteke sailkatzailearentzat atributu gehiegi eta konplexuegiak izatea. Beraz, hurrengo proba, Freeling-Ancora corpusean eginiko probaren atributu berdinak baliatuz egin genuen, hau da, oinarrizko atributuekin

	Doit.	**Est.**	F_1
Oin. ezaug. + lema + azpikat. + kate konplexua AVG	91,92	52,40	66,75
Oin. ezaug. + lema + azpikat. + kate konplexua LAST	89,93	49,30	63,69

Taula IV.4: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak, Ancora corpuseko atributu guztiak erabilita eta 10eko *epoch-zenbakia*. Garapen-corpusaren gainean egindako ebaluazioa.

(forma, kategoria eta katea). Corpusaren izaerak zeukan eragina ikusteko, beharrezkoa iruditu zitzaigun batean eta bestean atributu berdinak erabiltzea. IV.5 taulan ikus daitezke lortu genituen emaitzak.

	Doit.	**Est.**	F_1
Oin. ezaug. AVG	92,38	50,21	65,06
Oin. ezaug. LAST	79,22	51,85	62,68

Taula IV.5: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak, Ancora corpuseko oinarrizko atributuak erabilita eta 10eko *epoch-zenbakia*. Garapen-corpusaren gainean egindako ebaluazioa.

Emaitza hauek zalantzak areagotu zizkiguten. Informazio gutxiago eman-da ere emaitzak ez baitziren hobetu.

Emaitza hauek justifikatu nahian, corpusak sakonago aztertzen saiatu ginen. Alde batetik, emaitzetan ikusten denez, Ancora corpuseko estaldura Freeling-Ancora corpusekoa baino askoz baxuagoa da, nahiz eta doitasuna oso handia izan. Honek, nolabait, algoritmoa gutxi arriskatzen dela esan nahi du, agian daukan informazioa askotarikoa eta konplexuegia delako erabakiak hartzeko.

Bi corpusetako probetan, ia sei puntuko aldea egoteak eta automatikoki lortutako informazioarekin emaitza hobeak lortzeak (atributu berdinak erabiliz) asko harritu gintuen. Pentsatu genuen atributuen balioetan egon zitekeela gakoa. Hau egiaztatzeko asmoz, kateen informazioari zegozkion balioei erreparatu genien eta bi corpusetan desberdinak zirela ohartu ginen. Freeling-Ancora corpusean, katearen zutabeak 89 balio desberdin erabiltzen zituen, Freeling analizatzaile sintaktikoak hala itzulita. Ancora corpusean, ordea, 11 balio desberdin bakarrik zeuden.

Hemendik ondorio garbi bat atera genuen. Nahiz eta Freeling-Ancora

corpusean genuen informazioa, hasiera batean, Ancora corpusekoa baino txarragoa izan automatikoki lortua izateagatik, informazio aberatsagoa ematen zion, antza, *FR-Perceptron* ikasketa automatikoko programari. Ondorioz, Ancora corpuseko informazio falta hori beste atributuren batekin orekatu beharra ikusi genuen.

IV.3.4 Atributu linguistikoen konbinazio onena bilatuz

Aurreko atalean aipatu bezala, Ancora corpusean katearen informazioaren zutabean genuen informazio falta nolabait orekatu beharra genuen. Beraz, eskura geneuzkan atributuez baliatuz, hauen konbinazio onenaren bila hainbat proba egin genituen. Proiektuarekin bukatzen ari ginela, ohartu ginen BIO formatuan etiketazioa egitean interpretazio arazo bat izan zitekeela. Pentsatu genuen, hau izan zitekeela katearen informazioa gehitzean emaitzak horrenbeste okertzearen arrazoia, batez ere, kate konplexuarekin konbinatzean. Beraz, gaizki hartutako informazio hau zuzendu eta katearen informazioarekin egindako probak errepikatu genituen. IV.6 taulako datuak, BIO formatu egokiarekin lortuak izan dira. Hemendik aurrera *AVG* moduko emaitzak soilik aurkeztuko ditugu, *LAST* moduak baino emaitza hobea ematen dituelako.

	Doit.	**Est.**	F_1
Oin. ezaug.+lema+azpikategoria+kate konplexua. AVG	90,22	59,80	71,93
Oin. ezaug.+lema AVG	88,89	56,71	69,25
Oin. ezaug. AVG	88,73	55,18	68,05
forma+kategoria+kate konplexua AVG	87,71	68,50	76,93
forma+lema+kateg.+kate konplexua AVG	87,62	77,17	82,06
forma+lema+kateg.+azpikat.+kate konplexua AVG	88,39	78,99	**83,42**
forma+kateg.+azpikat.+kate konplexua AVG	88,75	78,09	83,08

Taula IV.6: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak, Ancora corpuseko atributuen konbinazio desberdinekin eta 10eko *epoch-zenbakia*. Garapen-corpusaren gainean egindako ebaluazioa zuzendutako kate atributuarekin.

Kasurik onenean 20 puntu hobetu genituen (F_1= % 83,42). Emaitza hauek Freeling-Ancora corpusarekin lortutakoak baino 12 puntu hobeak dira. Hemendik atera daitekeen lehen ondorioa, gauzak zuzen doazela da, hau da, Ancora corpuseko lehen emaitzak Freeling-Ancorakoak baino okerragoak

zirela ikustean arazo larriren bat zegoeneko susmoa izan genuen, baina azken emaitza hauek espero zitekeenaren barruan sartu gintuzten. Izan ere, eskuz etiketatutako informazio linguistikoarekin, hobeak behar lukete emaitzek. *FR-Perceptron* tresnaren funtzionamendua egokia izaten ari zela ere ziurtatu genuen, tresnaren tarteko emaitzak arakatuz.

Azken emaitza hauek analizatzean, nabarmena da *katearen informazio konplexua*ri dagokion atributuak emaitzetan eragin handia duela, emaitza onenak atributu hau gehitzean lortu baitira. Salbuespena, hala ere, Ancora corpuseko atributu guztiak erabiltzean aurkitzen dugu, non, nahiz eta *katearen informazio konplexua* atributua erabili, emaitzak oso txarrak baitziren. Antza denez, *katea* eta *katearen informazio konplexua* atributuen artean gatazka bat sortzen da. Nolabait, katearen informazioak eta honen gehigarri bezala ateratako informazioa ez datoz bat. Honek *FR-Perceptron* programari perpaus-muga ez jartzea eragiten dio estaldura 26 puntu okertzen.

Gauzak honela, Freeling-Ancora corpusaren ebaluazioan lortutako emaitzak hobetzea izan zen gure lana hemendik aurrera.

IV.3.5 Freeling-Ancora corpuseko emaitzak hobetu nahian

Freeling-Ancora corpusean emaitza hobeen bila ez genuen aukera askorik informazio linguistiko gehigarri gutxi genuelako, baina hasteko, *lema* atributua gehitu genion oinarrizko ezaugarriei, eta IV.7 taulan ikus daitezke emaitzak.

	Doit.	**Est.**	F_1
Oin. ezaug.	85,39	60,84	71,05
Oin. ezaug.+lema	85,59	62,19	**72,04**
Oin. ezaug.+azpikat.	85,47	61,03	71,21
Oin. ezaug.+lema+azpikat.	85,25	62,16	71,89

Taula IV.7: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak, Freeling-Ancora corpuseko oinarrizko atributuak,lema eta azpikategoria erabilita, 10eko *epoch-zenbakiarekin*. Garapen-corpusaren gainean egindako ebaluazioa.

Ebaluazio honekin, oinarrizko ezaugarriekin genuen emaitzetatik F_1 neurria puntu bat hobetzea lortu genuen, baina oraindik urruti geunden Ancora corpusarekin lortutako 83,4 puntuetatik, beraz nolabait informazio gehiago eman behar genion corpusari.

Informazio gehiago ematearen bilaketan, kategoriarako erabiltzen genuen karaktereaz ohartu ginen. Freeling analizatzaile sintaktikoak, kategoria definitzeko letra eta zenbaki ugariren konbinazioa duen kate bat itzultzen du, eta bere garaian, lehen karakterea bakarrik kontuan hartzea erabaki genuen, gainontzekoa ez baitzitzaigun esanguratsua iruditu. Esan beharra dago, Ancora jatorrizko corpusak ere, kate berdintsuak erabiltzen dituela tokenaren kategoria definitzeko garaian, eta hortik ere, lehenengo karakterea bakarrik hartu genuela. Ancora corpusean ordea, kategoriaz gain, azpikategoriaren informazioa genuen, eta honek informazio gehigarri bat ematen zuela konturatu ginen, nahiz eta Ancora corpusaren gainean eginiko ebaluazioek erakutsi diguten azpikategoria ez zela beste atributu batzuk bezain esanguratsua azken emaitzean.

Hala ere, informazio gehiago lortzeko beste modurik ez genuenez, Freeling-Ancora corpusean kategoria adierazteko karaktere bat erabili beharrean bi erabili genituen, bigarren karaktere honek azpikategoriaren informazioa ematen duelakoan.Honela Freeling-Ancora corpus berria sortu genuen, orain, kategoriarako bi karaktere erabiliz. IV.7 taulako azken bi ilaretan ikus daitezke ebaluazioen emaitzak.

Kategoriarako bi karaktere erabilita, emaitzak ez dira modu esanguratsuan aldatzen, beraz, aurreko ereduarekin geratzea erabaki genuen.

IV.3.6 Epoch zenbakia aldatzen

Interesgarria iruditu zitzaigun probak amaitutzat eman aurretik *epoch-zenbaki* desberdinarekin probatzea Arrieta (2010)en emaitzak zertxobait hobetzea lortu baitzen 15eko *epoch-zenbakia* erabiliz.

	Doit.	**Est.**	F_1
Ancora forma+lema+kat.+azpikat.+kate konplex.	88,31	79,74	83,81
Freeling-Ancora oin. inform.+lema	85,49	62,66	72,32

Taula IV.8: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak, Freeling-Ancora corpuseko eta Ancora corpuseko atributu aukeraketarik onena 15eko *epoch-zenbakiarekin*. Garapen-corpusaren gainean egindako ebaluazioa.

IV.8 taulan ikus daitekeenez, emaitzak ez dira aldatzen. Ondorioz, exekuzio denbora askoz gutxiago behar duenez, eta emaitzetan eraginik ez duenez,

10eko *epoch-zenbaki*arekin geratzea erabaki genuen *test*eko corpusaren gainean azken probak egiteko.

IV.3.7 Azken emaitzak, Test corpusean

Proba guztiak garapeneko corpusarekin ebaluatu genituen, eta bukatzeko, emaitzarik onenak lortzeko baliatutako *atributuak* erabiliz, test-corpusa erabili genuen azken ebaluazioak egiteko bi corpusekin.

	Doit.	**Est.**	F_1
Ancora garapen-corpusa	88,39	78,99	83,42
Ancora test-corpusa	88,20	77,99	82,78
Freeling-Ancora garapen-corpusa	85,59	62,19	72,04
Freeling-Ancora test-corpusa	84,99	61,51	71,37

Taula IV.9: Eskuz etiketatutako corpusa eta automatikoki etiketatutako corpusen azken emaitzak, garapen eta test corpusetan. Ancora corpusarekin erabilitako atributuak: forma, lema kategoria, azpikategoria eta katearen informazio konplexua. Freeling-Ancora corpusarekin erabilitako atributuak: forma, lema, kategoria eta katea. Guztietan 10eko *epoch zenbakia* erabili da.

Bai Ancora corpusa ebaluatzeko baita Freeling-Ancora ebaluatzeko ere, AVG modua erabili genuen, beti emaitza onenak ematen zizkigun modua baitzen.

IV.9 taulan ikus daitekeen moduan, kasu honetan test-corpusean lortzen ditugun emaitzak garapen-corpusekoak baino apur bat txikiagoak dira. Hala eta guztiz ere, desberdintasuna ez da handiegia, eta gaztelaniarako perpaus-identifikatzailearen neurria % 71 punturen bueltan finkatzeko balio digu proba honek.

V. KAPITULUA

Ondorioak

V.1 Azken ondorioak

Proiektu honek perpausen identifikazioak duen zailtasuna erakutsi digu. Egiaztatu dugu nola *FR-Perceptron* programari informazio gehiegi ematea emaitzen kalterako izan daitekeen, aldiz, informazio gutxi emanda baina hau oso aberatsa bada, emaitzak asko hobetu daitezke. Hori konprobatu ahal izan dugu *kate konplexua* atributuarekin.

Bestalde, esan daiteke, Freeling analizatzaile sintaktikoarekin sortutako corpusarekin ez ditugula uste bezain emaitza onak lortu, eta Ancora corpusarekin lortutako emaitzetatik nahiko urrun geratu garela. Honen arrazoia, Freeling analizatzaile sintaktikoan kokatu beharrean gaude, analizatzaileak emandako informazioa ez baita Ancora corpusean duguna bezain fidagarria, Ancora corpuseko informazioa eskuz etiketatua izan baita. V.1 taulan ikus daitezke azkeneko emaitzak, ingeleseko eta euskarako perpaus-identifikatzaileekin alderatuta. Kontuan izan behar da datuok interpretatzean, eskuz etiketatutako corpusarekin ikusi dugula gehienez ere % 82,78ko F_1 neurriraino irits gaitezkeela. Beraz, ondoriozta daiteke gaztelaniarako garatutako perpaus-identifikatzaile honen emaitzak hobetzea daudela. Esan beharra dago emaitza hauek nahiko harrigarriak iruditzen zaizkigula. Izan ere, gaztelaniak ingelesarekin dituen antzekotasunak euskarakoarekin lortzen ziren emaitzak hobetuko genituela iradokitzen ziguten. Ez da hala izan, eta badirudi Freeling analizatzaileak ematen digun kateei buruzko informazioa-

	F_1
Ingeleseko perpaus-identifikatzailea	84,36
Euskarako perpaus-identifikatzailea	77,24
Gaztelaniako perpaus-identifikatzailea	71,37

Taula V.1: Euskarako, ingeleseko eta gaztelaniako perpaus identifikatzaileen emaitzen arteko konparaketa, automatikoki analizatutako corpusekin eta *FR-Perceptron* algoritmoa erabiliz.

ren kalitateak daukala horren errua.

V.2 Etorkizunera begira

Proiektua amaituta, etorkizunera begira proiektu honetan oinarrituta beste lan batzuk egiteko aukera ikusten da.

Interesgarria litzateke entrenamendurako corpusaren tamaina handitzeak edo txikitzeak emaitzetan duen eragina aztertzea. Gure probetan CoNLL batzarretan erabilitako ingeleseko corpusaren tamaina erabili dugu, eta denbora faltagatik ezin izan dugu beste tamaina batzuekin probak egin.

Honez gain, Ancora corpusa beste analizatzaile sintaktiko batekin analizatzea interesgarria litzateke, Freelingek azken emaitzetan duen errua zenbaterainokoa den egiaztatzeko. Honekin batera, hartu dugun informazioaz gain beste atributu linguistiko batzuk ere hartzea ongi legoke, baina hau egin aurretik corpusa ongi analizatu beharko litzateke. Azken finean, interesgarria iruditu zitzaigun informazio guztia atera genuen.

Beste lan interesgarri bat corpusak lortzeko prozesua automatizatzea izango litzateke. Gu ahalik eta automatizatuen egiten saiatu gara, baina azkenean eskuzko lana egin behar izan dugu, Freeling-Ancora entrenamendurako corpusa prestatzeari hiruzpalau orduko eskuzko lana eskaini behar izan diogu.

Azkenik, ondo legoke *FR-Perceptron* programa konputagailu pertsonal batean instalatzeko beharrezkoak diren liburutegiak ongi definituak izatea. Gure proiektuan saiatu ginen ordenagailu pertsonalean instalatzen baina ezinezkoa izan zitzaigun, ez baikenezkien beharrezkoak ziren liburutegien bertsio zehatzak. Hala ere, esan beharra dago, IXA taldeko zerbitzarietan exekutatzeak hainbat abantaila dituela. *FR-Perceptron* programak ikasteko 4 egun inguru igarotzen baititu, eta zerbitzarietan exekutatzean abiaduran irabazten da.

V.3 Iritzi pertsonala

Proiektu hau hasiera batean pentsatu nuena baino interesgarriagoa suertatu zaidala aitortu beharra dut. Hasiera batean Hizkuntzaren Prozesamenduari buruz ezer gutxi nekien, baina aurrekariak aztertzen nituen ahala geroz eta interes handiagorekin heldu diot proiektuari eta amaieran gauzak sakonago aztertzeko gogoarekin geratu naiz.

Proiektuaren erronkarik garrantzitsuena inplementazio lana izan dela esango nuke. Nahiz eta programazio lana ez izan batere konplexua bere jatorrizko diseinuan, ez negoen batere ohituta JAVA programazio lengoaia erabiltzen, baina gustura hartu dut erronka hau, eta etorkizunerako esperientzia hau aberasgarria izango dela pentsatzen dut.

Bestalde, oso interesgarria iruditu zait *FR-Perceptron* bezalako herramin-ta bat erabili ahal izatea, oso programa indartsua iruditu zait eta ikasketa automatikoarekin gauza oso interesgarriak egitea daudela ohartu naiz, batez ere, Hizkuntzaren Prozesamenduaren alorrean.

Hizkuntzaren Prozesamenduari buruz ikasitako honek guztiak, etorkizunean alor honetan ikerketa gehiago egitearen aldeko atea zabalik uztera behartu nau, gai interesgarria iruditu baitzait eta etorkizunean garrantzia izan dezakeena.

VI. KAPITULUA

Eranskinak

VI.1 Bileren Aktak

1. Bileraren akta

Data eta ordua:	**2011ko urriaren 14ª (12:30-13:30)**
Helburua:	**KBParen aurkezpena eta plangintza orokorra.**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
Proiektuaren aurkezpen orokorra	Proiektuaren nondik norakoak azaldu dira eta irakurgai batzuk eman dira.
Hurrengo asteetako plangintza	Hurrengo asteetan jarraitu beharreko pausuen plangintza egin da. (1plangintza.docx)
Proiektuaren plangintza orokorra	Proiektuaren plangintza orokorra zehaztu da eta lehen mugarriak ezarri dira. (Plangintza_orokorra.docx)

Hurrengo bilera:	**2011ko urriaren 28a (12:30)**
Gaiak:	**1. Aurreko bileran adostutako eginbeharren berrikuspena** **2. Hurrengo asteetarako plangintza zehaztu**

2. Bileraren akta

Data eta ordua:	**2011ko urriaren 28a (12:30-13:30)**
Helburua:	**1. prototipoa ikusi eta hurrengo pausuak zehaztu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
1. prototipoaren emaitzak aztertu	Emaitzak egokiak diren ala ez konprobatu eta programaren funtzionamendua ikusi.
2. prototipoa prestatu	2. prototipoak izan behar dituen nondik norakoak zehaztu
XML dokumentuen morfologia aztertu	Tratatu beharreko XML fitxategien morfologia aztertu eta bertatik atera behar ditugun datuak nola errepresentatu adostu

Hurrengo bilera:	**2011ko azaroaren 11 (12:30)**
Gaiak:	**1. Bigarren prototipoaren emaitzak aztertu** **2. Hurrengo asteetarako plangintza zehaztu**

3. Bileraren akta

Data eta ordua:	**2011ko azaroaren 18a (12:30-13:30)**
Helburua:	**2. prototipoa ikusi eta hurrengo pausuak zehaztu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
2. prototipoaren emaitzak aztertu	Emaitzak egokiak diren ala ez konprobatu eta programaren funtzionamendua ikusi.
2. prototipoari falta zaiona adostu	2. prototipoari falta zaizkion funtzionalitateak definitu eta programa birfindu (azpiprogramak definitu)
Helburu dokumentua	Helburu dokumentua egiten hasi behar da, lana ez juntatzeko.

Hurrengo bilera:	**2011ko abenduaren 2a (12:30)**
Gaiak:	**1. Hirugarren prototipoa aztertu** **2. Hurrengo asteetarako plangintza zehaztu** **3. Landu beharreko dokumentazioa berrikusi**

4. Bileraren akta

Data eta ordua:	**2011ko abenduaren 2a (12:30-13:30)**
Helburua:	**3. prototipoa ikusi eta hurrengo pausuak zehaztu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
3. prototipoari falta zaiona adostu	3. prototipoaren detaileak hobetu, hutsuneak jarri, egiaztapen gehiago jarri.
Helburu dokumentua	Helburu dokumentua zer den ikusi eta gero, dokumentazioa aurreratzen hasi.
3. Protoripoaren bukaera aztertu	Irteera fitxategiaren amaieran gertatzen den arazoa aztertu eta konpondu

Hurrengo bilera:	**2011ko abenduaren 16a (12:30)**
Gaiak:	**1. Azken prototipoaren arazoak ikusi konpondu diren** **2. Egindako dokumentazioa aztertu** **3. Hurrengo asteetarako lana banatu**

5. Bileraren akta

Data eta ordua:	**2011ko abenduaren 16a (12:30-13:30)**
Helburua:	**Azken prototipoa ikusi eta dokumentazio lana banatu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Part of Speech	PoS atribututik balio digun informazio baliagarria bakarrik hartu eta beste zutabe berri batean jarri.
Dokumentazioa	Helburu dokumentua, aurrekariak etabar egiten hasi.
FR-Pertzeptroien programa instalatu	FR-Pertzeptroiekin lan egiteko beharr diren programak instalatu eta martxan jartzen saiatu.

Hurrengo bilera:	**2012ko urtarrilaren 13a (12:30)**
Gaiak:	**1. Egindako dokumentazioa ikusi.** **2. FR-Pertzeptroien programa instalatzean egon daitezkeen arazoak ikusi.** **3. Ikasketa automatikoarekin hasi.**

6. Bileraren akta

Data eta ordua:	**2012ko urtarrilaren 27a (12:30-13:30)**
Helburua:	**Ikasketa automatikoa martxan jarri**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
Dokumentazioa	Latex_en funtzionamendua ikasi dokumentazioa bertan egiten hasteko.
FR-Pertzeptroien programa instalatu	FR-Pertzeptroiekin lan egiteko behar den programa instalatzen saiatu.

Hurrengo bilera:	**2012ko otsailaren 3a (12:30)**
Gaiak:	**1. Ikasketa automatikoko programa instalatu den ala ez ikusi eta bestela unibertsitateko makinara konektatu probak bertan egiteko.** **2. Latexeko dokumentazioa ikusi.**

7. Bileraren akta

Data eta ordua:	**2012ko otsailaren 3a (12:30-13:30)**
Helburua:	**FR-Perceptron erabiltzen hasi**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
Testuaren etiketazioa	Gure testua eskuz etiketatua dago, beraz probak ere automatikoki etiketatuta egin beharko dira
Corpusa nola zatitu	Gutxi gora behera %70 entrenatzeko %15 garatzeko eta beste %15 testa egiteko

Hurrengo bilera:	**2012ko otsailaren 20a (12:30)**
Gaiak:	1. **LEXera pasatako dokumentazioa berrikusi** 2. **FR-Perceptron martxan jarri**

8. Bileraren akta

Data eta ordua:	**2012ko otsailaren 20a (12:30-13:30)**
Helburua:	**Freeling**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Testuaren etiketazioa	Testuak automatikoki etiketatu beharko dira freelingekin.
Corpusa lortu	Corpusaren testu soila lortu behar dugu, atributurik gabe, freelingi bidaltzeko
Freelingeko informazioa atera	Freelingek emandako etiketazio automatikoa gure formatura pasa beharko dugu eta zutabeka jarri

Hurrengo bilera:	**2012ko otsailaren 27a (12:30)**
Gaiak:	1. **Freelingekin ateratako informazioa ikusi.** 2. **FR-Perceptron martxan jarri.**

9. Bileraren akta

Data eta ordua:	**2012ko martxoaren 5a (12:30-13:30)**
Helburua:	**Freeling_eko arazoak konpondu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Komillak dituzten esaldiak	Komillak agertzen diren esaldiak, korpusetik kentzea erabaki dugu, Freelingek esaldi hauek tratatzean arazoak sortzen baitziren.
Preposizioa+artikulua arazoak	Freelingek preposizioak eta artikuluak banatuta jartzen ditu elkartuta egon behar dutenean. Konfigurazio fitxategia moldatu behar da.
Dokumentazioarekin jarraitu	Corpusaren osaketa azaldu dokumentu batean. PHD. Fr-Perceptron, Freeling eta Latexekin izandako arazoak arrisku posibleen atalean haipatu.

Hurrengo bilera:	**2012ko martxoaren 12a (12:30)**
Gaiak:	1. **FR-Perceptron martxan dabilen ikusi.** 2. **Korpusa osatuta ez badago, arazoak konpondu.** 3. **Dokumentazioa ikusi.**

10.Bileraren akta

Data eta ordua:	**2012ko martxoaren 23a (12:30-13:30)**
Helburua:	**Freeling_eko arazoak eta FR-Perceptronekin hasi**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Esaldi garbiketa	Freelingekin arazoak daudela ikusita, ahalik eta gehien konpontzen saiatuko gara, eta hortik aurrera, esaldi konfliktiboak korpusetik kentzea erabaki da.
Freelingeko Corpuseko hitzen PoS	Kategoria definitzeko, kategoria atributuko lehenengo letra bakarrik kontuan hartzea erabaki da, ikasketa prozesua eraginkorragoa izan dadin.
FR-Perceptron konfiguratu	FR-Perceptron Ancorako corpusarekin erabiltzeko konfiguratu da.

Hurrengo bilera:	**2012ko martxoaren 26a (12:30)**
Gaiak:	**1. FR-Perceptronen lehenengo proba egin** **2. Freelingekin dauden arazoak berrikusi**

11.Bileraren akta

Data eta ordua:	**2012ko martxoaren 26a (12:30-13:30)**
Helburua:	**FR-Perceptron lehen proba eta Freelingeko Corpusa**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
FR-Perceptron probatu	FR-Perceptronekin lehenengo emaitzak lortu dira eta bere funtzionamendua egokia dela ikusi da.
Freelingeko arazoak konpondu	Azken emaitza txarren ondorioz, freelingen konfigurazio zaharrera bueltatzea erabaki da eta azken probak egiten jarraitu Corpus egoki bat lortu harte.

Hurrengo bilera:	**2012ko apirilaren 2a (13:00)**
Gaiak:	**1. Dokumentazioko akatsen eztbaida** **2. Freelingeko Corpusaren egoera aztertu** **3. FR-Perceptron probatzen jarraitu**

12.Bileraren akta

Data eta ordua:	**2012ko apirilaren 2a (12:30-13:30)**
Helburua:	**Dokumentazioa eta Freeling**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
Dokumentazioa ikusi	Zuzendutako dokumentazioa berrikusi da
Freeling azken konfigurazioa	Freeling konfigurazio definitiboa adostu da, emaitza onenak ahalbidetzen dituena.

Hurrengo bilera:	**2012ko apirilaren 16a (12:30)**
Gaiak:	**1. Freelingeko Corpus osatuak aztertu** **2. FR-Perceptron probak hasi.**

13.Bileraren akta

Data eta ordua:	**2012ko apirilaren 16a (12:30-13:30)**
Helburua:	**Corpusekin amaitu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
Freelingeko Corpusa	Azken aldaketekin Freelingeko Corpusa handiagoa lortu daitekeela uste dugunez berriro egitea adostu da.
Ancora Corpusa txikitu	Ancora Corpusean agertzen diren esaldiak Freelingekoan agertzen direnak izateraino txikitzeko programatxo bat egitea erabaki da.

Hurrengo bilera:	**2012ko apirilaren 20a (11:30)**
Gaiak:	**1. Probak egiteko konfigurazio fitxategiak prestatu** **2. Probak martxan jarri**

14.Bileraren akta

Data eta ordua:	**2012ko apirilaren 20a (11:30-14:00)**
Helburua:	**FR-Clauser prestatu eta probak hasi**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
FR-Clauser konfiguratu	Konfigurazio fitxategiak egokitu dira Corpus mota bakoitzarekin modu egokian funtzionatzeko.
Lehen probak	FR-Clauser martxan jarri da lehenengo probekin.

Hurrengo bilera:	**2012ko maiatzaren 4a (12:30)**
Gaiak:	**1. Proben emaitzak aztertu.** **2. Exekuzioak nola doazen ikusi.**

15.Bileraren akta

Data eta ordua:	**2012ko maiatzaren 4a (11:30-14:00)**
Helburua:	**Lehen emaitzak aztertu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	**Iruzkinak**
Lehen emaitzak	Emaitzak aztertu ondoren, ustez baino txarragoak direla ikusi da, eta avg aukera onena dela adostu da emaitzak ikusita.
Hurrengo probak	Hurrengo probak Ancora eskuz etiketatutako datuekin egitea adostu da, erreferentziazko datu on batzuk izateko.

Hurrengo bilera:	**2012ko maiatzaren 11a (12:30)**
Gaiak:	**1. Proben emaitzak aztertu.** **2. Exekuzioak nola doazen ikusi.**

16.Bileraren akta

Data eta ordua:	**2012ko maiatzaren 11a (12:30-13:30)**
Helburua:	**Ancorako emaitzak aztertu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Emaitzen azterketa	Ancorako eskuz etiketatutako emaitzak aztertu ondoren Freelingekoak baino askoz txarragoak direla ohartu gara.
Emaitza txarren arrazoiaren bila	Gerta litekeen arazoaren bila, Freelingeko eta Ancorako chunk zutabeek har ditzaketen balio desberdinen zerrendak lortzea adostu dugu.
Beste proba berri bat	Emaitza txarren arrazoiaren bila, Ancora testuan beste proba berri bat egitea erabaki da, chunk zutabea alde batera utzita eta bere ordez chunk gehigarria erabiliz.
Arazo posible bat	Konfigurazio fitxategietan arazo bat egon daitekeen susmoa daukagu, beraz aldaketa txiki batzuk egin ondoren (C eta P) probak berriro egingo dira eta emaitza aztertu.

Hurrengo bilera:	**2012ko maiatzaren 25a (12:30)**
Gaiak:	**1. Proben emaitzak aztertu.** **2. Debuggeatu.**

17.Bileraren akta

Data eta ordua:	**2012ko maiatzaren 25a (12:30-13:30)**
Helburua:	**Debuggeatu eta emaitzak aztertu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Debuggeatu	FR-Perceptron programa debuggeatu ondoren, guztiak modu egokian funtzionatzen duela ikusi dugu.
Emaitzen azterketa	Chunk-complex atributuarekin probak egitea erabaki dugu, emaitza hobeak lortzeko asmoarekin

Hurrengo bilera:	**2012ko ekainaren 22a (12:30)**
Gaiak:	**1. Proben emaitzak aztertu.** **2. Dokumentazioa errepasatu**

18.Bileraren akta

Data eta ordua:	**2012ko ekainaren 22a (12:30-13:30)**
Helburua:	**Azken probak eta dokumentazioa**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Dokumentazioaren zuzenketa	Dokumentazioa aztertu da eta egin beharreko zuzenketak adostu dira.
Emaitzen azterketa	BIO zuzendu behar dela adostu da, eta proba berriak martxan jarri.
Epoch 15	Epoch 15 zenbakiarekin probak egitea erabaki da

Hurrengo bilera:	**2012ko uztailaren 5a (12:30)**
Gaiak:	**1. Proben emaitzak aztertu.** **2. Dokumentazioa zuzendu.**

19.Bileraren akta

Data eta ordua:	**2012ko uztailaren 5a (11:30-13:00)**
Helburua:	**Epoch emaitzak aztertu eta dokumentazioa zuzendu**
Partaideak:	**Bertol Arrieta (Proiektuaren zuzendaria); Andoni Ibirriaga (Ikaslea)**

Gaia	Iruzkinak
Dokumentazioaren zuzenketa	Dokumentazioa aztertu da eta egin beharreko zuzenketak adostu dira.
Emaitzen azterketa	Martxan dauden proben emaitzak aztertu dira.

Hurrengo bilera:	**2012ko uztailaren 10a (12:30)**
Gaiak:	**1. Proben emaitzak aztertu.** **2. Dokumentazioaren azken zuzenketak.**

Bibliografia

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K., eta Zabala I. Hitz-hurrenkeraren azterketa masiboa corpusean. Barne-txostena, EHU, 2003.

Arrieta B. *Azaleko sintaxiaren tratamendua ikasketa automatikoko teknikken bidez: euskarako kateen eta perpausen identifikazioa eta bere erabilera koma-zuzentzaile batean*. Doktoretza-tesia, Informatika fakultatea : EHU-UPV, 2010.

Black E., Abney S., Flickenger D., Gdaniec C., Grisham R., Harrison P., Hindle D., Ingria R., Jelinek F., Klavans J., Liberman M., Marcus M., Roukos S., Santorini B., eta Strzalkowski T. A procedure for quantitatively comparing the syntactic coverage of English grammars. *Proceedings of DARPA Workshop on Speech and Natural Language*, 1991.

Carreras X. *Learning and Inference in Phrase Recognition: A Filtering-Ranking Architecture using Perceptron*. Doktoretza-tesia, Polytechnic University of Catalunya, 2005.

Carreras X., Màrquez L., eta Castro J. Filtering-ranking perceptron learning for partial parsing. *Machine Learning Journal, Special Issue on Learning in Speech and Language Technologies*, 60(1-3):41–71, 2005.

Collins M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *EMNLP*, 2002.

Freund Y. eta Schapire R.E. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

Màrquez L. Aprendizaje automático y procesamiento del lenguaje natural. *Tratamiento del lenguaje natural*, page 207, 2002.

Sang E.T.K. eta Buchholz S. Introduction to the conll-2000 shared task: Chunking. *Proceedings of Computational Natural Language Learning*, Lisbon (Portugal), 2000.

Sang E.T.K. eta Déjean H. Introduction to the conll-2001 shared task: Clause identification. *Proceedings of Computational Natural Language Learning*, Toulouse (France), 2001.